



FACULTAD DE INGENIERÍA Y CIENCIAS APLICADAS

DESARROLLO DE UNA APLICACIÓN WEB DE RED SOCIAL PARA
ESTIMULAR LOS VÍNCULOS AFECTIVOS ENTRE MASCOTAS Y SUS
DUEÑOS.

AUTORA

Valeria Nicole Zaldumbide Burbano

AÑO

2018



FACULTAD DE INGENIERÍAS Y CIENCIAS APLICADAS

DESARROLLO DE UNA APLICACIÓN WEB DE RED SOCIAL PARA
ESTIMULAR LOS VÍNCULOS AFECTIVOS ENTRE MASCOTAS Y SUS
DUEÑOS.

Trabajo de Titulación presentado en conformidad con los requisitos
establecidos para optar por el título de Ingeniera en Sistemas de Computación
e Informática

Profesor Guía

MSc. Pedro Manuel Nogales Cobas

Autora

Valeria Nicole Zaldumbide Burbano

Año

2018

DECLARACIÓN DEL PROFESOR GUÍA

"Declaro haber dirigido el trabajo, Desarrollo de una aplicación web de red social para estimular los vínculos afectivos entre mascotas y sus dueños, a través de reuniones periódicas con el estudiante Valeria Nicole Zaldumbide Burbano, en el semestre 2018-2 orientando sus conocimientos y competencias para un eficiente desarrollo del tema escogido y dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación".

Pedro Manuel Nogales Cobas.

Master en Gestión de Proyectos Informáticos

1756760284

DECLARACIÓN DEL PROFESOR CORRECTOR

"Declaro haber revisado este trabajo, Desarrollo de una aplicación web de red social para estimular los vínculos afectivos entre mascotas y sus dueños, de Valeria Nicole Zaldumbide Burbano, en el semestre 2018-2, dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación".

Anita Elizabeth Yáñez Torres.

Magíster en Gerencia Empresarial

1802462216

DECLARACIÓN DE AUTORÍA DEL ESTUDIANTE

“Declaro que este trabajo es original, de mi autoría, que se han citado las fuentes correspondientes y que en su ejecución se respetaron las disposiciones legales que protegen los derechos de autor vigentes.”

Valeria Nicole Zaldumbide Burbano

1723737068

AGRADECIMIENTOS

A mi familia, maestros y amigos.

DEDICATORIA

Quiero dedicar este trabajo a mis padres, ya que, sin ellos y su apoyo incansable, esto no hubiera sido posible. A Addison, Meepo, Bruno y a la Sra.Puma, por su amor animal.

RESUMEN

El amor animal ha brindado beneficios a los seres humanos a lo largo de la historia. La estimulación de estos vínculos afectivos entre personas y mascotas brinda ventajas no solo físicas, emocionales y espirituales, sino también evolutivas. Existen en el mercado actualmente varias aplicaciones para los animales de compañía, pero no con el enfoque y el objetivo de la solución aquí presente. *Pet Lovers*, es una aplicación web de red social para estimular los vínculos afectivos entre mascotas y sus dueños. En el presente documento se muestra su desarrollo, es decir, su marco teórico, el desarrollo de la solución y la validación de esta. Gracias a la aplicación las personas podrán interactuar con otras y con las mascotas de estas, creando ambientes propicios para utilización de los beneficios que estos vínculos representan. La solución fue realizada con el framework Scrum, comprende una aplicación web programada en PHP, con la ayuda del framework Laravel con base de datos MySQL y combinado con tecnologías para la mensajería instantánea como Pusher y Vue.js. Utilizando el patrón Modelo, Vista y Controlador.

ABSTRACT

Animal love has provided benefits to human beings throughout history. The stimulation of these affective bonds between people and pets offers advantages not only physical, emotional and spiritual, but also evolutionary. Currently there are several applications for pets, but not with the focus and the objective of the solution here. *Pet Lovers* is a social network web application to stimulate affective bonds between pets and their owners. In the present document its development is shown, that is, its theoretical framework, the development of the solution and the validation of it. Thanks to the application, people will be able to interact with others and with their pets, creating favorable environments for using the benefits that these links represent. The solution was made with the Scrum framework, comprising a web application programmed in PHP, with the help of Laravel framework with MySQL database and combined with technologies for instant messaging such as Pusher and Vue.js. Using the Model, View and Controller pattern.

ÍNDICE

Introducción.....	1
Antecedentes	1
Alcance	3
Justificación	4
Objetivo General	5
Objetivos Específicos.....	5
1. CAPÍTULO I. Fundamentación teórica	6
1.1 Introducción	6
1.2 Relación humano-animal.....	6
1.2.1 Convivencia.....	6
1.2.2 Beneficios.....	8
1.2.3 Terapia con animales	10
1.3 Herramientas y aplicaciones similares	11
1.4 Metodología de desarrollo.....	13
1.4.1 Definición.....	13
1.4.2 Historia	14
1.4.3 Roles, Artefactos y Actividades	14
1.4.4 Ciclo de vida de Scrum.....	16

1.4.5 Medición y estimación ágil.....	17
1.5 Tecnologías	18
1.5.1 Framework de desarrollo	18
1.5.2 Lenguaje de programación.....	20
1.5.3 Base de Datos.....	20
1.5.3 Otras herramientas	22
2. CAPÍTULO II. Propuesta de Solución.....	23
2.1 Introducción	23
2.2 <i>Product Backlog</i>	23
2.3 <i>Sprints Backlogs</i>	26
2.4 Gráficos	29
2.5 Diagrama Entidad Relación	30
2.6 Modelo Vista Controlador	31
2.7 Desarrollo.....	34
2.7.1 Migraciones y Base de Datos.....	34
2.7.2 Clases	37
2.7.3 Controladores.....	40
2.7.4 Rutas	42
2.7.5 Vistas.....	43

2.8 Interfaces de usuario	47
3.CAPÍTULO III. Validación	53
3.1 Introducción	53
3.2 Pruebas de aceptación del cliente	53
3.3 Pruebas Unitarias.....	65
4.Conclusiones y Recomendaciones.....	76
4.1 Conclusiones.....	76
4.2 Recomendaciones	77
Referencias.....	78

ÍNDICE DE FIGURAS

<i>Figura 1.</i> Ciclo de vida de Scrum.....	17
<i>Figura 2.</i> Arquitectura de Pusher.....	22
<i>Figura 3.</i> Gráfico final de Burn up.....	29
<i>Figura 4.</i> Gráfico de final de Burn down del Sprint 1	29
<i>Figura 5.</i> Gráfico de final de Burn down del Sprint 2	30
<i>Figura 6.</i> Diagrama entidad relación de la solución	31
<i>Figura 7.</i> Modelo de la solución	32
<i>Figura 8.</i> Vista de la solución	32
<i>Figura 9.</i> Controlador de la solución	33
<i>Figura 10.</i> MVC Laravel de la solución	33
<i>Figura 11.</i> Parte del código para el desarrollo de la tabla de usuarios.	34
<i>Figura 12.</i> Parte del código para el desarrollo de la tabla de mascotas.	34
<i>Figura 13.</i> Parte del código para el desarrollo de la tabla de interacciones.....	35
<i>Figura 14.</i> Parte del código para el desarrollo de la tabla de matches.	35
<i>Figura 15.</i> Parte del código para el desarrollo de la tabla de mensajes.	36
<i>Figura 16.</i> Ejecución del comando “ <i>php artisan migrate</i> ” para la migración.	36
<i>Figura 17.</i> Base de datos de la solución.	37
<i>Figura 18.</i> Parte del código para el desarrollo de la clase usuario.....	38
<i>Figura 19.</i> Parte del código para el desarrollo de la clase mascota.....	38
<i>Figura 20.</i> Parte del código para el desarrollo de la clase interacción.	39
<i>Figura 21.</i> Parte del código para el desarrollo de la clase match.....	39
<i>Figura 22.</i> Parte del código para el desarrollo de la clase mensaje.....	40
<i>Figura 23.</i> Parte del código para el desarrollo del controlador de mascotas. ...	40

<i>Figura 24.</i> Parte del código para el desarrollo del controlador de matches.....	41
<i>Figura 25.</i> Parte del código para el desarrollo del controlador de	41
<i>Figura 26.</i> Parte del código para el desarrollo del controlador de historial.	42
<i>Figura 27.</i> Parte del código para el desarrollo del controlador de mensajes. ...	42
<i>Figura 28.</i> Parte del código para el desarrollo de las rutas de la solución.....	43
<i>Figura 29.</i> Parte del código para el desarrollo de la vista de bienvenida.....	43
<i>Figura 30.</i> Parte del código para el desarrollo de la vista de inicio.	44
<i>Figura 31.</i> Parte del código para el desarrollo de la vista de creación de mascota.	44
<i>Figura 32.</i> Parte del código para el desarrollo de la vista de interacciones y matches.	45
<i>Figura 33.</i> Parte del código para el desarrollo de la vista de notificaciones.	45
<i>Figura 34.</i> Parte del código para el desarrollo de la vista de historial.	46
<i>Figura 35.</i> Parte del código para el desarrollo de la vista de mensajes.....	46
<i>Figura 36.</i> Interfaz de bienvenida.	47
<i>Figura 37.</i> Interfaz de registro.	47
<i>Figura 38.</i> Interfaz de ingreso.	48
<i>Figura 39.</i> Interfaz de inicio.	48
<i>Figura 40.</i> Interfaz de ingreso de mascota.	49
<i>Figura 41.</i> Interfaz de mascotas.	49
<i>Figura 42.</i> Interfaz de mostrar mascota.....	50
<i>Figura 43.</i> Interfaz de editar mascota.	50
<i>Figura 44.</i> Interfaz de patitas para las interacciones y matches.....	51
<i>Figura 45.</i> Interfaz de notificaciones.....	51

<i>Figura 46.</i> Interfaz de historial.....	52
<i>Figura 47.</i> Interfaz de mensajes.	52
<i>Figura 48.</i> Interfaz del canal de chat.	52
<i>Figura 49.</i> Interfaz de mascotas del usuario.	53
<i>Figura 50.</i> Demostración del paso 1 parte primera.....	55
<i>Figura 51.</i> Demostración del paso 1 parte segunda.	56
<i>Figura 52.</i> Demostración del paso 2 parte primera.....	56
<i>Figura 53.</i> Demostración del paso 2 parte segunda.	57
<i>Figura 54.</i> Demostración del paso 3 parte primera.....	57
<i>Figura 55.</i> Demostración del paso 3 parte segunda.	58
<i>Figura 56.</i> Demostración del paso 4.....	58
<i>Figura 57.</i> Demostración del paso 5 parte primera.....	59
<i>Figura 58.</i> Demostración del paso 5 parte segunda.	59
<i>Figura 59.</i> Demostración del paso 6 parte primera.....	60
<i>Figura 60.</i> Demostración del paso 6 parte segunda.	60
<i>Figura 61.</i> Demostración del paso 6 parte tercera.....	61
<i>Figura 62.</i> Demostración del paso 7 parte primera.....	61
<i>Figura 63.</i> Demostración del paso 7 parte segunda.	62
<i>Figura 64.</i> Demostración del paso 7 parte tercera.....	62
<i>Figura 65.</i> Demostración del paso 8 parte primera.....	63
<i>Figura 66.</i> Demostración del paso 8 parte segunda.	63
<i>Figura 67.</i> Demostración del paso 8 parte tercera.....	63
<i>Figura 68.</i> Demostración del paso 9 parte primera.....	64
<i>Figura 69.</i> Demostración del paso 9 parte segunda.	64

<i>Figura 70.</i> Demostración del paso 10.....	64
<i>Figura 71.</i> Demostración del paso 11 parte 1.	65
<i>Figura 72.</i> Demostración del paso 11 parte 2.	65
<i>Figura 73.</i> Demostración del caso 1 parte primera.	67
<i>Figura 74.</i> Demostración del caso 1 parte segunda.	67
<i>Figura 75.</i> Demostración del caso 2 parte primera.	68
<i>Figura 76.</i> Demostración del caso 2 parte segunda.	68
<i>Figura 77.</i> Demostración del caso 2 parte tercera.	69
<i>Figura 78.</i> Demostración del caso 3 parte primera.	69
<i>Figura 79.</i> Demostración del caso 3 parte segunda.	70
<i>Figura 80.</i> Demostración del caso 3 parte segunda.	70
<i>Figura 81.</i> Demostración del caso 3 parte tercera.	71
<i>Figura 82.</i> Demostración del caso 3 parte cuarta.	71
<i>Figura 83.</i> Demostración del caso 4.	72
<i>Figura 84.</i> Demostración del caso 5.	72
<i>Figura 85.</i> Demostración del caso 6.	72
<i>Figura 86.</i> Demostración del caso 7.	73
<i>Figura 87.</i> Demostración del caso 8.	73
<i>Figura 88.</i> Demostración del caso 9.	73
<i>Figura 89.</i> Demostración del caso 10 parte primera.	73
<i>Figura 90.</i> Demostración del caso 10 parte segunda.	74
<i>Figura 91.</i> Demostración del caso 11 parte primera.	74
<i>Figura 92.</i> Demostración del caso 11 parte segunda.	74
<i>Figura 93.</i> Demostración del caso 11 parte tercera.	75

<i>Figura 94.</i> Demostración del caso 11 parte cuarta.	75
<i>Figura 95.</i> Demostración del caso 12.	75

Introducción

El presente trabajo muestra el desarrollo del proyecto web del tipo red social que busca apoyar la estimulación de los vínculos afectivos entre mascotas y sus dueños. Se describirá los antecedentes, justificación, alcance y objetivos. La fundamentación teórica cuenta con una investigación sobre la relación de humanos y animales, la metodología de desarrollo, las herramientas y aplicaciones similares que se pueden encontrar en la actualidad en el mercado, y las tecnologías a usar. La propuesta de solución describe la elaboración de la aplicación web la cual apoyará la estimulación de los vínculos afectivos entre los usuarios y sus animales de compañía mediante mensajes instantáneos como consecuencia de un sistema de gusto o afición, además se validará la propuesta de solución y los resultados de esta. Para finalizar se elaborará las conclusiones generales, las recomendaciones y referencias.

Antecedentes

Se define a la mascota como animal que se mantiene en el hogar y se lo trata amablemente. Es decir, son animales de compañía que las personas atesoran para su propio beneficio y disfrute, velando por el bienestar del animal (Cambridge University Press, 2018).

El trato del ser humano hacia los animales domésticos es completamente diferente al trato que se tiene hacia los animales de crianza, los destinados a ganado, animales para el transporte o de laboratorio, estos brindan ganancias económicas y como alimento a la humanidad. En cambio, los animales de compañía proveen al ser humano beneficios tales como amor, ternura, cariño, afecto por citar algunos.

Desde hace siglos, el ser humano ha tenido compañeros animales. Los perros fueron uno de los primeros animales que migraron con los humanos hacia todos los continentes y los felinos han acompañado al hombre desde tiempos antes de Cristo.

Otros animales se han unido como acompañantes en la vida del ser humano a lo largo de la historia de este y los ha denominado “mascotas”. En la actualidad, el ser humano se preocupa por sus mascotas considerándoles no solo animales sino amigos y buscando lo mejor para estos (Gómez, Atehortua y Orozco, 2007).

Por ejemplo, algunas mascotas disfrutan de beneficios de tiendas especializadas en donde se puede encontrar la venta de artículos básicos y de lujo para su consumo, servicios estéticos, médicos, recreativos, educativos, viviendas lujosas, juguetes, entre otros, que los seres humanos han creado solo para la satisfacción y el cuidado de sus compañeros animales.

Es decir, la relación que el ser humano posee con los animales de compañía continúa estrechándose y desarrollándose. No solo influenciados por la convivencia sino también por el afecto emocional que ha ido teniendo la humanidad a lo largo de los siglos (Tucker, 2017).

El libro “*The Animals Among Us: How Pets Make Us Human*” muestra que existe mucha historia y cultura en la manera en que se expresa el deseo de cuidar de los animales, pero que en resumen tiende a ser un instinto humano. Un instinto que tiene beneficios evolutivos, los cuales son descritos en el libro “*Pets in the Family: An Evolutionary Perspective*”. Estos libros afirman que las personas al ser seres sociales buscan de manera continua la relación con los demás incluso con las mascotas propias y ajenas. Si los individuos carecen de este apoyo social son más vulnerables a enfermedades e infecciones (Bradshaw, 2017) (Serpell y Paul, 2011).

En el artículo “*Biology, Culture, and the Origins of Pet-Keeping*” se asegura que las personas mantienen mascotas porque otras personas lo hacen, es decir, porque es "socialmente contagioso", siendo este el resultado de captar las señales sociales de los demás. En otras palabras, el mantenimiento de mascotas es un meme, una tendencia que se ve continuamente reforzada por su propia popularidad (Herzog, 2014).

De acuerdo a toda esta información anteriormente descrita, es complicado saber con exactitud por qué los seres humanos poseen animales de compañía o mascotas ya que es la unión de varios factores. Lo que sí es palpable son los lazos afectivos que existen entre ellos. Entonces la pregunta a responder es la siguiente: “¿Por qué los humanos aman tanto a sus mascotas?”.

Clive Wynne , director del Canine Science Collaboratory en la Universidad Estatal de Arizona ha dedicado su carrera al estudio del comportamiento animal y la relación evolutiva entre animales y personas. Wynne responde que la pregunta es realmente increíble y que a lo largo de la historia se sabe cómo se llegó a amar a los animales de compañía pero que, a pesar de todas las teorías presentadas para explicar el amor por las mascotas, aún faltan más y mejores datos para determinar la razón del porqué.

La fascinación que el ser humano experimenta por sus mascotas es algo innegable y de igual manera los lazos afectivos que estos comparten (Kaplan, 2017).

Tomando todo lo dicho anteriormente se evidencia la necesidad y la importancia de las mascotas en la vida del ser humano y por qué se debe reforzar y fomentar los vínculos afectivos entre humanos y animales de compañía. Por lo tanto, se propone el desarrollo de la aplicación *Pet Lovers* la cual será una red social para estimular los vínculos afectivos entre mascotas y sus dueños.

Alcance

La solución se basa en el apoyo para estimular vínculos afectivos entre personas y animales de compañía por medio del desarrollo una aplicación web.

El objetivo de la aplicación es conocer a las mascotas y sus respectivos dueños por medio una patita (botón de “like”) que dará el usuario demostrando su interés o afición y produciendo una interacción que podría convertirse en una unión que denominaremos “*match*”.

Para lograr esto el usuario será capaz de ingresar a sus mascotas al sistema (cualquier tipo de mascota), crearles un perfil y buscar a otras mascotas en la aplicación. Por ejemplo, si un usuario “A” da patita a una mascota de un usuario “B” se formará una interacción. Si “B” devuelve la patita a una mascota de “A” entonces se formará una interacción completa también llamada *match* y se procederá a desbloquear un canal de mensajería instantánea para ambos.

El patrón de arquitectura a usar es el modelo - vista - controlador también llamado MVC. Todos los datos serán guardados utilizando la base de datos MySQL y será realizado con el framework Laravel en lenguaje PHP 5. La interfaz de usuario será realizada con HTML 5, Angular 1 y Bootstrap 4.

Para la comunicación entre usuarios se utilizarán Vue.js el cual es un marco de JavaScript progresivo y Pusher que es un servicio en línea que utiliza websockets para mensajería en tiempo real.

La metodología por utilizar será SCRUM, el cual es framework que posee ventajas de monitoreo para avances de Sprint. Lo que diferencia a esta metodología es que indica el trabajo que falta por realizar, así como la velocidad del equipo en cuanto al cumplimiento de las historias de usuario comprometidas en el Sprint.

Justificación

Con respecto a la población ecuatoriana que podría estar interesada en la solución, se tienen varias estadísticas publicadas de los diarios locales y avaladas por el Instituto Nacional de Estadística y Censos (INEC).

Según el diario el comercio, dueños de mascotas dedican más de dos horas semanales a sus cuidados. La noticia “Dueños de mascotas dedican más de dos horas semanales a sus cuidados” redacta según datos del INEC las personas dedican en promedio dos horas y 21 minutos a la semana, al cuidado de sus mascotas (EL COMERCIO, 2014).

Esto significa que la población ecuatoriana está interesada en el cuidado y tiempo de dedicación que les da a sus mascotas.

Otro dato del diario el ciudadano en su noticia “53 millones de dólares genera el mercado de los alimentos para mascotas” redacta que el INEC publicó un reporte que demuestra la elaboración de 10 millones de kilos de alimentos para perros y que, en promedio, se elaboran 167 772 juguetes para mascotas, el cual supera la cifra de 73 000 en ventas. (El ciudadano, 2017)

Gracias a estas cifras se puede afirmar que una gran parte de la población ecuatoriana posee una mascota y que invierte tiempo en sus cuidados. Lo que afirma que se está en busca de la conexión humano-animal.

Lo que se necesita es no solo mantener esta conexión, sino que hacerla más fuerte entre las mascotas y sus dueños, buscando un acercamiento social entre las personas que tienen mascotas y las propias mascotas en sí, con la oportunidad de crear aún más conexiones.

Es por eso por lo que se decide que la aplicación abarque todo tipo de mascota para promover la unión. También es requerido una herramienta donde los que son dueños de varias mascotas no deban estar cambiando de aplicación en aplicación, sino que encuentren en una, todo lo que necesitan para poder conectar con sus mascotas y las mascotas de otros.

Objetivo General

Desarrollar una aplicación web que permita a través de “Match” estimular los vínculos afectivos entre personas y animales de compañía.

Objetivos Específicos

- Realizar la fundamentación teórica para determinar el ambiente en el que será utilizada la aplicación.
- Desarrollar la aplicación web *Pet Lovers* que propicie los vínculos afectivos entre personas y animales a través de “match”.

- Validar la propuesta de solución con la aplicación de pruebas de software.

1. CAPÍTULO I. Fundamentación teórica

1.1 Introducción

En este capítulo, se desarrolla la fundamentación teórica del proyecto en base al planteamiento del problema en cuestión. Se trata exclusivamente los asuntos o temas relacionados con el problema, buscando plasmar con facilidad los conocimientos e información descrita para que la vinculación de las ideas, teorías y procedimientos sean no solamente sencillos de comprender sino altamente útiles y prácticos.

Se expondrá la relación humano-animal, su significado, su explicación, su utilidad y lo que realmente representa para el ser humano. Luego, se procederá a describir soluciones parecidas o planteadas anteriormente en el mercado, así como también variantes a la misma. Se describirá la metodología con la que se desarrollará la solución y para finalizar se expondrán las herramientas de software que serán utilizadas para resolver la problemática planteada en la introducción del presente documento.

Esta sección es esencial para comprender el problema como tal y por qué es importante. Con el desarrollo de este se podrá respaldar la solución propuesta, la cual consiste en el desarrollo de una web de red social para estimular los vínculos afectivos entre mascotas y sus dueños.

1.2 Relación humano-animal

1.2.1 Convivencia

El foco de la relación entre humanos y animales en la cual se desarrollará el tema será únicamente como vínculo dueño-mascota, es decir, la convivencia de animales domésticos en la sociedad humana.

Los seres humanos muestran una unión especial con sus mascotas que no hallan con otras, los animales tienen una forma de aceptación sin juicios, no condicionan su amor, perdonan fácilmente. Es por esto por lo que las relaciones

con los animales de compañía implican ser menos difíciles que las relaciones con otras personas (Pichot y Coulter, 2007, pp.12).

La relación entre personas y animales de compañía es un hábito que puede ser casi tan viejo como nuestra especie. Para poder entender este vínculo, Bradshaw recurre a cuatro características de la naturaleza humana que provoca la unión hacia sus mascotas.

La primera llama “cute responde” traducido como linda respuesta se basa en el cuidado de mamíferos hacia sus crías, es decir, la activación que este provoca en el cerebro al ver un ser mucho más pequeño y con características denominadas “adorables”. Aunque existe diferencia entre los bebés humanos y las crías de animales mamíferos, poseen en la actualidad la misma característica de provocar lindura. En algún momento de la historia, el sentimiento que solo se tenía hacia los bebés humanos bifurcó hacia las crías de los animales de tal manera que llegó a convertirse en un sistema que responde a captar el atractivo de los animales como “tierno y lindo”.

La segunda característica es el poder de la caricia, mecanismo importante que el ser humano desarrolla antes que el lenguaje. No solo existe el gusto del tacto hacia las mascotas sino también las personas encuentran relajación en el mismo. Gracias a ese sistema de caricias se libera dopamina y endorfina en el cerebro, el cual permite la unión de lazos afectivos entre las personas y los animales domésticos, aparte de ser una manera de comunicación bidireccional.

La tercera es el disfrute que tienen los seres humanos de nutrir a otras especies, rasgo que únicamente se presenta en humanos y es gracias a esta característica que las personas pudieron lograr desarrollar la capacidad de domesticar animales. Es esta característica la que hace que juguetes del tipo Tamagotchi (aparato electrónico que despliega una mascota virtual) y AIBO (mascotas robóticas) puedan producir sentimientos parecidos en sus poseedores.

La participación de la oxitocina (hormona del amor materno) en el vínculo, es la razón por la cual el ser humano invierte tiempo, energía y recursos hacia sus mascotas (Caba, 2003) (Bradshaw,2017).

Especialmente las mujeres son las que más presentan esta característica, se cree que por diferencias de género en base fisiológica. Por ejemplo, la disminución de la soledad que los animales de compañía aportan a las mujeres de mediana edad, pero no tanto a los hombres, pueden provenir de respuestas hormonales que experimentan las mujeres al atender de sus mascotas diariamente. La retroalimentación positiva que esto les ofrece es esencial para mantener y profundizar la relación.

La cuarta es la capacidad del hombre por antropomorfizar (conceder forma o cualidades humanas), e intentar entrar en la mente de los animales que se encuentran a su alrededor. Heredado en parte de mecanismos cerebrales que examinan el comportamiento de otros humanos, este comportamiento universal surgió desde tiempos antiguos en que entender al animal salvaje que ponía en peligro al cazador era una característica que aseguraba su supervivencia o su muerte. Ahora, la antropomorfización es la que provoca que las personas afirmen que sus mascotas son parte de su familia, al ser tan cercanas a las mismas saben cuándo su compañero tiene hambre, pide una caricia o requiere de una atención en específico (RAE, 2018).

Estas características fueron evolucionándose, logrando que los humanos y los animales formaran un vínculo dueño-mascota a lo largo de los años hasta convertirse en como lo conocemos hoy día (Bradshaw,2017).

1.2.2 Beneficios

Aparte de todas las características antes mencionadas que posee el hombre y las razones que lo unen a los animales de compañía, también existen ventajas de esta unión.

Muchos estudios e investigaciones acreditan que los animales de compañía proveen de varios beneficios para el ser humano. Aún si el fin de la persona para tener a su lado una mascota es solamente en el ámbito social, este animal brinda mucho más a la persona que solo compañía.

Gran cantidad de personas optan por este tipo de compañía, no solo porque les parece más genuina, sino que afirman que en un mundo en donde las relaciones

con las personas pueden ser demandantes, estresantes y llenas de conflicto, los animales proveen al ser humano un descanso al solo buscar la compañía de su dueño.

Es esencial para los niños el tener animales de compañía, ya que estos ayudan a desarrollar responsabilidades, capacidades, autoestima, confianza, y empatía. La mascota brinda al niño no solo de un aliado de juegos, si no de un confidente y alguien con quien practicar sus habilidades sociales, del mismo modo que, da una de las primeras experiencias de muerte y la pérdida (Pichot y Coulter, 2007).

Para la gente mayor también representa una ayuda significativa e importante, El COMERCIO en su publicación “Estudio demuestra beneficios para los adultos mayores de tener una mascota” redacta que la interacción con las mascotas les ayuda con la presión arterial, facilita la relajación y les brinda compañía, entre otros. Es decir, que poseer una mascota les ayuda no solo a su actividad física y mental sino también con la soledad que experimentan en esa etapa de sus vidas (ELCOMERCIO, s.f).

Clewley afirma que el amor animal de las mascotas brinda a mucha gente la razón para despertarse en las mañanas dándoles un verdadero poder de curación. Los animales de compañía no solamente brindan esto sino también pueden ayudar a las personas a mejorar, esto es posible gracias a que muchas mascotas protegen a los miembros de su familia adoptiva. Los muchos beneficios que las mascotas proveen con su compañía son avalados por varios estudios científicos.

Entre las muchas ventajas que las mascotas proporcionan a las personas, se pueden apreciar en la ayuda que estos brindan tanto de manera mental como emocional.

De acuerdo a PAWS (“Pets for people”, organización líder en terapia de mascotas), las mascotas brindan a los seres humanos:

- Reducen la presión sanguínea.
- Curan sentimientos de soledad.
- Ayudan con distintos desordenes.

- Apoyan de manera emocional y física.
- Proveen confort.
- Ayudan a evitar el aislamiento.
- Mejoran la salud del corazón.
- Liberan endorfinas en el cerebro.
- Reducen el dolor físico
- Evitan la depresión

Las mascotas poseen muchas ventajas y ayudan a la mayoría de las personas de diferentes maneras. No se abasteca a decir suficiente respecto a las numerosas ayudas tanto físicas, mentales, emocionales y espirituales que ofrecen los animales de compañía, pero se puede asegurar con firmeza que la fuerza de curación que brindan es una de las más efectivas que existen (Clewley, 2016).

1.2.3 Terapia con animales

Gracias a los beneficios antes mencionados emerge lo que hoy conocemos como terapia con animales también llamada TAAC. Varios estudios científicos avalan la utilidad y beneficio que significa el tratamiento con animales de compañía categorizándola como “buena medicina”.

TAAC tiene la ventaja de ser utilizada con una gran variedad de personas. Por ejemplo, para los niños y adolescentes es utilizado para la estimulación de habilidades psicomotoras y lenguaje. El tratamiento es dirigido hacia niños y adolescentes con habilidades especiales tanto físicas como psicológicas con trastornos emocionales, jóvenes con tendencia a la delincuencia, o con problemas de autismo, alumnos con necesidades de educación especial o retraso mental límite (Zamarra, 2002).

Otro tratamiento también que brinda la fuerza del poder animal se puede ver reflejado también en los hospitales de niños con cáncer, los cuales instantáneamente sonrían aun cuando su lucha en la vida es difícil. La alegría y el disfrute de estos niños son la mejor medicina antidepresiva que puede existir.

Las personas mayores tienen también tratamientos terapéuticos con animales de compañía ya que como estos poseen beneficios tales como ayudar con la soledad, depresión, entre otros. Para mucho de ellos son un remplazo o sustituto de los hijos que no tuvieron o ya no están. Los animales de compañía les permiten ser independientes sin sentirse solos, estimulando los sentidos de vista, oído, olfato y tacto, incrementando su actividad física y desarrollo muscular.

TAAC también es utilizada en unidades psiquiátricas, registrando disminución de suicidios en adultos con depresión y disminución de tiempo de estadía en hospitales psiquiátricos. Siendo también positivo con los pacientes de retraso mental (Villalta V., & Ochoa S., 2007).

En las prisiones también se emplean tratamientos con los animales de compañía. Se ha comprado sus beneficios y aprovechándolos para reducir la violencia y comportamientos antisociales, suicidios, adicción a sustancias tóxicas, mejor relación entre los reclusos y hacia los trabajadores de la institución. TAAC tiene un gran impacto facilitando la recuperación de estas personas y ayudando en su reinserción en la sociedad con su poder animal.

En todas las áreas antes mencionadas en las cuales se aplica TAAC, se tiene en común la búsqueda de comunicación entre humano y animal con el objetivo del beneficio entre ambos. Zamorra afirma que cuando existen lágrimas, dolor, sufrimiento y penas profundas los animales no huyen, como suelen hacerlo los seres humanos, cerrándose al dolor ajeno (Zamorra, 2002).

1.3 Herramientas y aplicaciones similares

En el mercado actual existen herramientas y aplicaciones similares. Entre ellas tenemos Mascotea, Social Animals, Yummypets, Twindog, Beppupy, DateMyPet, Facepets y VivaPets.

Mascotea es una página web española la cual apunta a ser un universo para las mascotas, es decir es tanto una red social, compra y vende artículos, blogs, información de mascotas, veterinarias cercanas y viajes con mascotas (Mascotea, 2018).

Social Animals es una aplicación disponible en Apple store y Google play, salió a principios de este año 2018, consiste en mascotas y anuncios, conversaciones con los dueños de las mascotas, búsqueda por proximidad y calendario de eventos (Social Animals, 2018).

Yummypets es una página en línea que se posiciona como una comunidad en internet para las mascotas y sus dueños. Se siguen a las mascotas de los miembros, tienen una revista en línea, información completa de razas, tipos de mascotas y foro (Yummypets, 2016).

Twindog es una aplicación que al igual que “Tinder”, permite la unión mediante un “match” por parte de los usuarios para luego conseguir el dialogo entre ellos. Solo es destinado para canes. Esa disponible en Apple store y Google play (Twindog,2018).

Bepuppy es una aplicación web italiana de red social para dueños de las mascotas. Aquí pueden agregar a su mascota conversar con otras y realizar publicaciones. La mayoría de las publicaciones se encuentran en italiano (Bepuppy, s.f).

DateMyPet es una red social en la cual se centra en la unión de personas que tiene el mismo gusto por las mascotas. Por el momento solo está disponible en inglés y para solo siete países con la misma habla (DateMyPet, 2018).

Facepets es una aplicación web que aborda todo el universo de las mascotas, con tienda en línea, adopción, y búsqueda de pareja de estas. Por el momento se encuentra en producción beta (Facepets, s.f).

VivaPets es una comunidad en línea llena de información de mascotas y miembros que participan en foros. Está disponible en tres idiomas y posee una gran variedad de fotos en sus galerías (VivaPets, 2018).

Todas estas aplicaciones son similares, pero no iguales a la solución propuesta en este documento.

1.4 Metodología de desarrollo

La metodología de desarrollo que se utilizará para la realización de la solución será Scrum. Se seleccionó esta metodología por ser un modelo ágil del desarrollo de software.

1.4.1 Definición

“Scrum es un marco de trabajo (framework) para la gestión de productos, proyectos y servicios complejos, que facilita un desarrollo mantenido e incremental”.(Monte, 2017)

Scrum es un modelo de desarrollo ágil caracterizado por: Adoptar una estrategia de desarrollo incremental, en lugar de la planificación y ejecución completa del producto. Basar la calidad del resultado más en el conocimiento tácito de las personas en equipos autoorganizados, que en la calidad de los procesos empleados. Solapamiento de las diferentes fases del desarrollo, en lugar de realizar una tras otra en un ciclo secuencial o de cascada (ScrumManage, 2014).

El control práctico de procesos para la gestión de sistemas adaptativos complejos es en lo que se basa Scrum. Los tres cimientos son: transparencia, inspección y adaptación.

Scrum define premisas generales para el proyecto las cuales son las siguientes: satisfacción del cliente, afinidad ante el cambio de requisitos, trabajo orientado en el proyecto, servicio o producto, definido como desarrollo sostenible, basado en cooperación frecuente, comunicación inmediata y continua entre individuos, personas motivadas frente a individuos dirigidos, disposición a la excelencia, simplicidad, equipos autoorganizados y adaptables. En cuanto a sus valores existe compromiso, enfoque, abierto, respeto y coraje.

Gracias a sus premisas, valores y características como las entregas incrementales, la aportación continua de valor y control de tiempo entre otras, la estudiante en cuestión se decidió por este marco de referencia (Montes, 2017).

1.4.2 Historia

El modelo como tal fue reconocido y especificado por Ikujiro Nonaka e Hirotaka Takeuchi a inicios de 1980, al examinar cómo desplegaban las principales empresas de manufactura tecnológica los nuevos productos.

En el estudio de Nonaka y Takeuchi se compararon nuevas formas de trabajo en equipo, teniendo presente la alineación de scrum de los jugadores de Rugby, gracias a esto se acuñó el término “scrum” para referirse a ella (Nonaka y Takeuchi, 1986).

Aunque se sabe que el origen de esta forma de trabajo nació en compañías de productos basados en tecnología, se notó que también es adecuada para proyectos con requisitos inconsistentes, necesitando rapidez y flexibilidad en los mismos, con escenarios habituales en el desarrollo de software.

En 1995, Ken Schwaber mostró “Scrum Development Process” en OOPSLA 95 (Object-Oriented Programming Systems & Applications conference), en el cual se exponía un marco de pautas para el desarrollo de software, basándose en bases de scrum, y que él había puesto en práctica en el desarrollo de Delphi (ScrumManage, 2014).

1.4.3 Roles, Artefactos y Actividades

Scrum propone:

- Roles:
 - *Product Owner* (PP): También denominado propietario del producto es aquel que determina las prioridades.
 - *Scrum Team* (E): También llamado equipo de desarrollo, son aquellos que construyen el producto
 - *Scrum master* (SM): Administra y proporciona la realización de las reglas de Scrum.
 - *Stakeholders* (I): Denominados "partes interesadas" o "interesados" son el resto de los implicados son encargados de asesorar y observar.
- Artefactos:

- *Product Backlog* (Pila del producto): Lista priorizada relacionada con los requisitos de usuario. Tiene una visión inicial que luego crece y evoluciona durante el desarrollo.
- *Sprint Backlog* (Pila del Sprint): Lista de requerimientos involucrados por el equipo para el Sprint con el suficiente detalle para su realización.
- Incremento: Resultado como parte del producto de un sprint en condición de ser utilizado en pruebas, documentación o codificación.
- Actividades:
 - *Sprint Planning* (Planificación del sprint): Es una reunión definida en una jornada de trabajo como máximo, el *product owner* (PP) explica las prioridades, se estima el esfuerzo a ser realizado basándose en los requisitos prioritarios y se realiza la pila del script, el equipo (E) define en una frase el fin del sprint.
 - Sprint: Es el ciclo básico de scrum, es decir, el marco estándar. Su duración recomendada es inferior a un mes y no mayor de seis semanas. Se designa sprint a cada iteración de trabajo que causa una pieza del producto terminada, funcionalmente y operativa (incremento).
 - *Daily Scrum* (Scrum diario): Es una reunión de quince minutos como máximo y el equipo es el responsable del mismo (E). En estas reuniones se define “Lo realizado ayer”, “Lo que se realizará hoy” y “Problemas obtenidos”. Al finalizar se renueva la pila del sprint.
 - *Sprint Review* (Revisión del Sprint): Tiene como objetivo que sea informativa, su máximo recomendado son cuatro horas, en él, se muestran sugerencias y se notifica el próximo sprint.
 - Sprint Retrospective (Retrospectiva del sprint): El equipo tiene como obligación autoanalizarse en su manera de trabajar para así encontrar fortalezas (las cuales se refuerzan) y debilidades (se realiza un plan de mejora).

Los roles marcan cómo se dividirán las responsabilidades en un proyecto. Tenemos por un lado al equipo de trabajo (*Scrum team*) que es el encargado del desarrollo del producto. El propietario del producto (*product owner*) que representa al negocio y el *Scrum master* que son parte del organismo de gobierno del proyecto. Y, por último, los concedores del negocio y la necesidad que son los usuarios clave o interesados (*stakeholders*).

Los artefactos son parte del equipo de herramientas y gracias a estos el equipo de trabajo puede ejecutar y controlar el proyecto. Se encuentran basados principalmente en listas de control, pilas del producto, pilas del sprint e incremento. Lo beneficioso es la adaptabilidad del framework, ya que, no llegan a ser mandatorios u obligatorios el uso de los artefactos, permitiendo a si ajustarlos de acuerdo a las necesidades del equipo de trabajo y al escenario del proyecto.

Para finalizar, las actividades son parte de los tiempos de realización del proyecto. La partición de este en ciclos autoorganizados y evaluables, con el fin de facilitar en todo momento un enfoque sobre el producto que se está desarrollando.

1.4.4 Ciclo de vida de Scrum

Un proyecto Scrum comienza con el *product owner*, este, es el encargado de crear e informar el *product backlog* (PB) el cual es formado por historias de usuario las cuales muestran un aspecto o funcionalidad que requiere el producto. Luego el *product owner* llamado también propietario del producto, prioriza las historias de usuario dándoles orden.

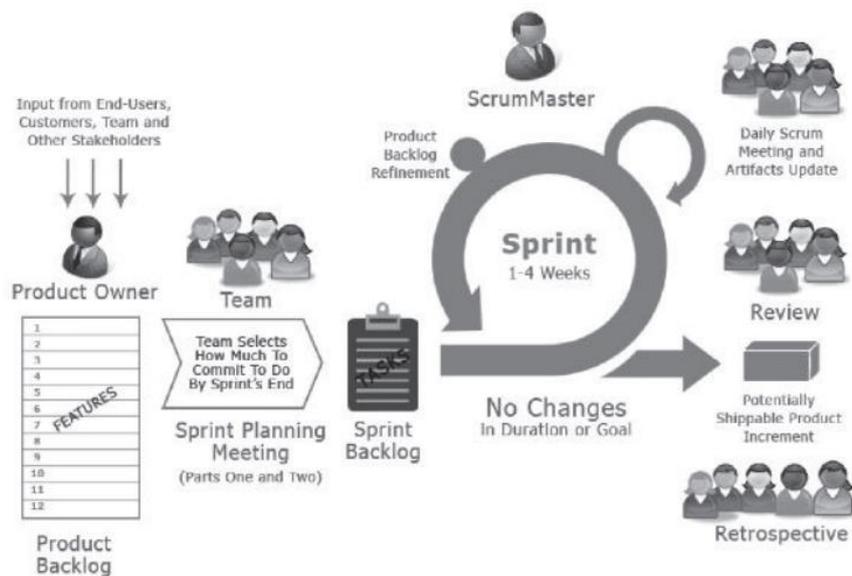


Figura 1. Ciclo de vida de Scrum.

Adaptado de Monte, 2017.

Para cada uno de los *sprints* del proyecto, el *Scrum team* decide qué funcionalidades del *product backlog* se unen en el *sprint backlog* que se encuentre actual y, entonces, el *development team* divide las historias de usuario en tareas y se realiza la estimación de esfuerzo para cada tarea. Una vez hecho esto, y mientras dure el *sprint*, se realizarán las reuniones diarias también llamadas *daily meetings* en las cuales el equipo coincidirá de manera interna y progresará en la realización de las tareas.

El proceso tiene que terminar con una reunión de exposición de resultados al usuario (*sprint review*) el cual será el encargado de aceptar o rechazar el producto creado. Por último, el *development team* y el *Scrum master* se unirán para una reunión (*scrum retrospective*) en la cual identificarán áreas de mejora en el proceso interno e iniciarán un nuevo *sprint* (Monte, 2017).

1.4.5 Medición y estimación ágil

Es necesario un proceso de medición para poder comprender a nivel de organización el rendimiento, la satisfacción del cliente, la eficiencia. A nivel de proyecto tenemos el esfuerzo, costo y tiempo. A nivel de desarrollo tenemos la cantidad de fallas, complejidad en el diseño, la disponibilidad y las horas de

esfuerzo. Al ser la información la materia prima para las decisiones, al estar cuantificada, proporciona criterios objetivos de gestión y seguimiento.

Las mediciones ayudarán a hacer sprints mejor planeados, garantizarán los tiempos en la ejecución de las historias de usuario o tareas del proyecto, permitiendo el control del estado del proyecto.

El sistema de estimaciones que brinda Scrum es de dos gráficos, *Burn up* o gráfico del producto y *Burn-down* o gráfico de avance.

- *Burn up* (Gráfico de producto): El encargado es el propietario del producto y es una herramienta de planificación, la cual, muestra de forma visual la evolución previsible del proyecto, el eje de las ordenadas mide el esfuerzo estimado para construir las historias de la lista de producto, el tiempo es medido en Sprint o en tiempo Real. Si se interceptan sus ejes cartesianos se determina la velocidad en desarrollo.
- *Burn-down* (Gráfico de avance): El equipo es el encargado de actualizar este gráfico en las reuniones del seguimiento del script. Permite el monitoreo y ritmo del avance como tal, detectando posibles desviaciones de lo previsto anteriormente. En el eje "Y" se encuentra el trabajo faltante y en el eje "X" los días del Sprint. Es estratégico para la medición del esfuerzo faltante y la realización del seguimiento diario (ScrumManage, 2014).

1.5 Tecnologías

1.5.1 Framework de desarrollo

Un marco de trabajo o *framework* (marco de trabajo) se utiliza en varios ámbitos del desarrollo de sistemas de software como por ejemplo en el desarrollo de aplicaciones médicas, de visión por computador, para el desarrollo de juegos, entre otros.

El termino framework hace referencia a una estructura de software compuesta de componentes intercambiables y personalizables para el desarrollo de una aplicación. Es decir, un framework se puede definir como una aplicación genérica

en estado incompleto pero configurable a la que es posible agregar piezas para construir una aplicación en concreto.

Los objetivos de la utilización de estos marcos de trabajo son el de acelerar el proceso de desarrollo, promover buenas prácticas de desarrollo como el uso de patrones y reutilizar código existente.

Entonces, un framework web es por lo tanto un conjunto de componentes que son parte de un modelo reutilizable, facilitando el desarrollo de un sistema web.

Para poder comprender el funcionamiento de los marcos de trabajo es crucial conocer el patrón Modelo – Vista – Controlador (MVC). MVC distribuye la aplicación en tres modelos separados. El primero es el modelo que representa los datos de la aplicación y las reglas de negocio. El segundo es la vista que representa el conjunto de vistas gráficas o formularios de entrada y salida de información. El tercero es el controlador que representa a los controladores del sistema, los cuales procesan las peticiones de los usuarios, controlando y siguiendo el flujo de ejecución del sistema (Gutiérrez, s.f).

De entre todos los frameworks que existen en la actualidad se seleccionó el framework Laravel para el desarrollo del proyecto por su extensa documentación y alta popularidad.

Laravel, es un framework o marco de trabajo de código abierto, creado en el 2011 y en continua mejora, se utiliza para el desarrollo de aplicaciones web y servicios web con PHP 5. Su ética es desarrollar código PHP de forma elegante y simple.

Laravel usa el paradigma de programación orientado a objetos (POO), admite el uso del patrón MVC, así como también el mapeo objeto-relacional (ORM). Al tener la mayoría de sus partes formadas por dependencias, especialmente de Symfony que es un framework diseñado para optimizar el desarrollo de las aplicaciones web basado en el patrón MVC, involucra que su desarrollo dependa del desarrollo de sus dependencias. A la vez, Laravel, plantea el uso de '*Routes with closures*', que reemplaza al MVC tradicional con el fin de realizar un código

más claro. A pesar de eso también permite el uso de MVC tradicional. (Sierra, Acosta, Ariza y Salas, 2013)

1.5.2 Lenguaje de programación

El lenguaje que se utilizará en la solución será PHP, el cual es un lenguaje de programación de código abierto interpretado por parte del lado del servidor y su acrónimo es *Hypertext Preprocessor*. En la actualidad es muy conocido y utilizado para el desarrollo web teniendo como opción la incrustación en HTML.

PHP es un código que se encuentra contenido entre dos etiquetas puestas al inicio y final, las cuales son las siguientes: “<?php” y “?>”. Estas etiquetas permiten la entrada y salida del “modo PHP”.

Lo que marca la diferencia a PHP con otros lenguajes de programación web, es que es un lenguaje orientado a objetos, admite programación concurrente, dispone de clases de objetos para la generación de interfaces gráficas de usuario, tiene prestaciones multimedia, lenguaje denominado simple, robusto y seguro (Cobo,2005).

La gran simplicidad para el principiante es una de sus principales características de PHP, brindando además opciones avanzadas para programadores que se desarrollan en un ámbito profesional.

PHP sugiere a sus usuarios que no tengan temor de leer la larga lista de características que posee, más bien aseguran que en pocas horas de lectura ya se logrará iniciar escribiendo los primeros scripts (PHP, 2018).

1.5.3 Base de Datos

El sistema de administración de bases de datos que se eligió para este proyecto es MySQL, el cual se encuentra basado en un lenguaje de consulta estructurado también llamado SQL. Es distribuido, respaldado y desarrollado por la corporación Oracle.

MySQL utiliza SQL, el cual es un lenguaje estandarizado, llegando a ser el más común para acceder a las bases de datos. En base a su medio de programación, se puede especificar código SQL directamente o en código escrito en otro idioma

o usar una interfaz de programación de aplicaciones API concreta que oculte la sintaxis de SQL.

MySQL al ser una base de datos, forma un sistema de administración o gestión de datos. Estos datos se guardan de manera estructurada en una base de datos. Por ejemplo, una base de datos o también llamada BD, puede llegar a ser cualquier cosa, desde una sencilla lista de supermercado hasta guardar gran cantidad de imágenes o albergar una cantidad significativa de información de una importante y gran empresa.

Para poder manejar los datos, administrarlos y gestionarlos se necesita un sistema de gestión de base de datos, como es el servidor MySQL el cual puede agregar, acceder y procesar datos almacenados.

MySQL es una DB que se caracteriza por almacenar datos en tablas separadas, evitando así poner todos los datos en un almacén. Las estructuras de la base de datos están establecidas en ficheros físicos mejorados en velocidad. Los objetos en bases de datos como tablas, vistas, filas y columnas, es decir, el modelo lógico, brinda un ambiente de programación flexible.

Al ser MySQL de código abierto, significa que es posible que cualquier usuario use y cambie el software.

El servidor de base de datos MySQL se caracteriza por su rapidez, escalabilidad, confiabilidad y su facilidad de uso, pudiendo ejecutarse en una computadora sencilla o compleja, de la misma forma que las con otras aplicaciones, servidores web, etc.

El servidor MySQL sigue en desarrollo y actualmente ofrece un paquete de funciones útiles y altamente ricas en funcionalidad. Su conectividad, velocidad y seguridad hacen que sea apropiado para permitir el acceso de bases de datos en Internet.

MySQL trabaja en sistemas cliente – servidor o en sistemas integrados. El primero que es el que se utilizará en el proyecto, consiste en la utilización de un servidor SQL multiproceso que admite diferentes back-ends, herramientas

administrativas, varios programas de cliente y bibliotecas y una extensa escala de interfaces de programación de aplicaciones (MySQL, 2018).

1.5.3 Otras herramientas

Ya que la solución necesita poseer un módulo de mensajes en tiempo real por lo cual se logrará realizar esto utilizando la tecnología Pusher y Vue.js.

Pusher es una solución en tiempo real altamente escalable establecida en 2010 viniendo a ser una API (interfaz de programación de aplicaciones) de servicio en línea que contiene la implementación de tecnología que permite la comunicación bidireccional entre un cliente y un host (*websockets*) y ejecuta un servidor de websockets propio recurriendo a la conectividad basada en HTTP, facilitando la actualización del número de mensajes enviados y nuevas conexiones en tiempo real como se aprecia en la figura 2. (Gauchat, 2013) (Pusher, 2018).



Figura 2. Arquitectura de Pusher.

Adaptado de Pusher, 2018

Vue.js es un marco progresivo para edificar interfaces de usuario, esbozado desde cero, es incremental y adaptable. La biblioteca central está orientada solo en la capa de visualización, siendo fácil de elegir e integrar con otras bibliotecas o proyectos existentes.

También, Vue.js es idóneo al impulsar sofisticadas aplicaciones de una sola página cuando se utiliza en combinación con herramientas modernas y bibliotecas de soporte (Vue, 2018).

2. CAPÍTULO II. Propuesta de Solución

2.1 Introducción

En este capítulo, se procederá a desarrollar la propuesta de la solución planteada con el objetivo de estimular los vínculos afectivos entre personas y animales de compañía para el beneficio mutuo de ambos.

Se demostrará la utilización del framework Scum para el desarrollo de la solución. Se documentará la realización del *product backlog*, así como los *sprint backlogs* obtenidos del mismo. Se procederá a desarrollar las gráficas *Burn up* y *Brun down* e ir actualizándolas a lo largo del desarrollo del proyecto. En cada incremento se procederá a mostrar el resultado obtenido, así como la retrospectiva de este. Se definirá el diagrama entidad-relación. Por último, se desplegarán una serie de vistas funcionales donde se mostrará la estructura del patrón MVC y el código de programación.

2.2 Product Backlog

A continuación, se presentará el product backlog de la solución el cual muestra la lista completa con las historias de usuario requeridas para desarrollar la solución en una tabla denominada tabla 1.

La tabla 1 posee cuatro columnas:

1. La primera denominada "Id", es el identificador único del trabajo o funcionalidad.
2. La segunda columna denominada "Prioridad", es el campo que muestra la diferencia de importancia.

3. La tercera columna definida como "Descripción" se encuentra en tercera persona y es la funcionalidad o requisito denominado "historias de usuario".
4. La cuarta columna "estimación" propone un número el cual representa la estimación del esfuerzo necesario.

En total se tiene 54 puntos de historia (P.H) que es la sumatoria de los números en la columna "estimación". Cabe destacar que la estimación de este proyecto se realizó basado en Puntos de Historia.

Tabla 1.

Product backlog de la solución.

Id	Prioridad	Descripción	Estimación
1	Baja	El usuario no autorizado necesita ir a la página de inicio y leer una descripción de lo que hace la aplicación web para sugerirle la creación de una cuenta.	1
2	Alta	El usuario no autorizado necesita crear una cuenta nueva para poder acceder a los beneficios de la aplicación web.	2
3	Media	El usuario no autorizado necesita iniciar sesión para ingresar a la aplicación.	2
4	Media	El usuario autorizado necesita cerrar sesión para finalizar su ingreso a la aplicación.	2
5	Alta	El usuario autorizado necesita ingresar a sus mascotas para poder crearles los perfiles respectivos.	5

6	Alta	El usuario autorizado necesita buscar a las mascotas ingresadas en el sistema para poder observar las opciones disponibles.	2
7	Media	El usuario autorizado necesita poder filtrar la búsqueda por el tipo de mascota para la facilidad de búsqueda.	5
8	Alta	El usuario autorizado necesita dar "like" o "patita" al perfil de la mascota de un usuario para poder demostrar su interés.	4
9	Alta	El usuario autorizado necesita comunicarse con los usuarios que le devolvieron el "like" o "patita" para poder entablar una conexión de "match".	8
10	Media	El usuario autorizado necesita ver, editar o eliminar a sus mascotas ingresadas para poder tener control sobre las mismas.	5
11	Baja	El usuario autorizado necesita ver un historial de los diferentes "like" o "patitas" que ha dado a otros usuarios para poder asegurarse de su actividad en la aplicación.	3
12	Baja	El usuario autorizado necesita que se le notifique si un usuario ha dado "like" o "patita" a una de sus mascotas para poder estar informado.	3

13	Alta	El usuario autorizado necesita ver en “Mensajes” que usuarios se encuentran activos y cuales inactivos.	6
14	Media	El usuario autorizado necesita ver en sus notificaciones un enlace en el nombre del usuario que dio “patita” para poder ver que mascotas están relacionadas con ese usuario.	6

El product Backlog presentado se encuentra ya actualizado a la última versión, después de la reunión de planificación para definir la versión 5.2 de la solución que se propone.

2.3 Sprints Backlogs

Los *sprints backlogs* son listas que descomponen las funcionalidades descritas en el *product backlog* en actividades necesarias para un incremento, por lo tanto, se basó tanto en la prioridad como en la función del incremento para dividir en 2 sprints de un mes cada uno.

1. Primer Sprint: Concentra los siguientes identificadores de las historias de usuario: 2,5,6,8,9,13. Con un total de 27 P.H (tabla 2)
2. Segundo Sprint: Concentra los siguientes identificadores de las historias de usuario: 1,3,4,7,10,11,12,14. Con un total de 27 P.H (tabla 2)

Los scripts mostrados a continuación describen su estado como completado ya que fueron siendo realizados a medida que la solución fue desarrollada. Comenzaron todos en estado de pendiente, pasaron por un estado denominado en proceso y cuando se culminó su desarrollo pasaron a tener el estado de completado (terminado).

En la parte de las fechas de los sprints se tienen los puntos de esfuerzo real realizado por cada una de las historias de usuario planteadas.

Tabla 2.

Sprint Backlog del Sprint 1

Sprint 1						
Id	Tarea	Estado	02/04/18	09/04/18	16/04/18	23/04/18
2	El usuario no autorizado necesita crear una cuenta nueva para poder acceder a los beneficios de la aplicación w eb.	Completo	3	0	0	0
5	El usuario autorizado necesita ingresar a sus mascotas para poder crearles los perfiles respectivos.	Completo	5	0	0	0
6	El usuario autorizado necesita buscar a las mascotas ingresadas en el sistema para poder observar las opciones disponibles.	Completo	2	1	0	0
8	El usuario autorizado necesita dar patita al perfil de la mascota de un usuario para poder demostrar su interés.	Completo	3	1	2	0
9	El usuario autorizado necesita comunicarse con los usuarios que le devolvieron "patita" para poder entablar una conexión.	Completo	0	4	5	2
13	El usuario autorizado necesita ver en "Mensajes" que usuarios se encuentran activos y cuales inactivos.	Completo	0	0	0	6

Tabla 3.

Sprint Backlog del Sprint 2

Sprint 2							
Id	Tarea	Estado	30/04/18	07/05/18	14/05/18	21/05/18	28/05/18
1	El usuario no autorizado necesita ir a página inicio y leer una descripción de la aplicación web para sugerirle la creación de una cuenta.	Completo	1	1	0	0	0
3	El usuario no autorizado necesita iniciar sesión para ingresar a la aplicación.	Completo	2	0	0	1	0
4	El usuario autorizado necesita cerrar sesión para finalizar su ingreso a la aplicación.	Completo	4	1	3	0	0
7	El usuario autorizado necesita poder filtrar la búsqueda por el tipo de mascota para la facilidad de búsqueda.	Completo	1	2	2	0	0
10	El usuario autorizado necesita ver, editar o eliminar a sus mascotas ingresadas para poder tener control sobre las mismas.	Completo	1	1	1	3	0
11	El usuario autorizado necesita ver un historial de los diferentes "patitas" que ha dado a otros usuarios para poder asegurarse de mi actividad en la aplicación.	Completo	2	1	1	2	0
12	El usuario autorizado necesita que se le notifique si un usuario ha dado "patita" a una de sus mascotas para poder estar informado	Completo	3	2	1	0	0
14	El usuario autorizado necesita ver en sus notificaciones un enlace en el nombre del usuario que dio "patita" para poder ver que mascotas están relacionadas con ese usuario	Completo	0	0	0	1	5

2.4 Gráficos

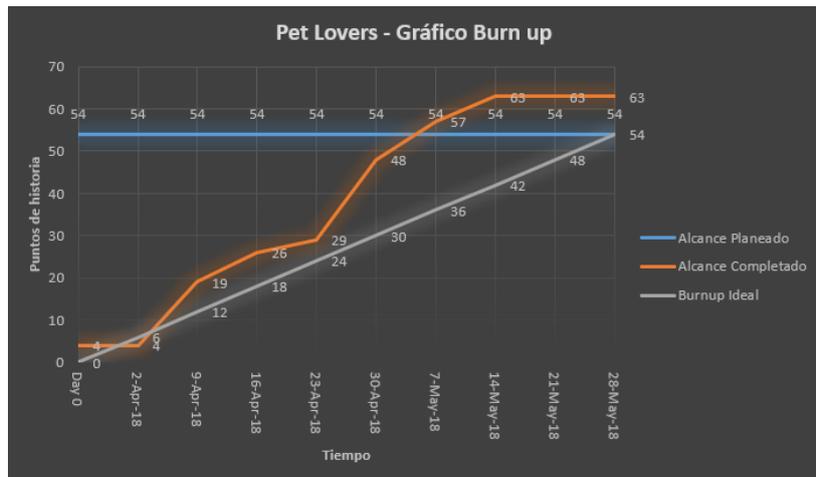


Figura 3. Gráfico final de Burn up

De acuerdo con la figura 3, se puede apreciar que el burn up ideal no fue conseguido y que la estimación planeada fue más baja que el alcance completado.

La velocidad fue optimista, pero sobrepasó el esfuerzo estimado en un principio, traspasando los 54 puntos de historia llegando a 63 puntos. Es decir, hubo un fallo de estimación por 16,66%.

La gráfica no fue planteada para representar un plan estable o estructurado sino como representación de un product backlog previsto y estimado.

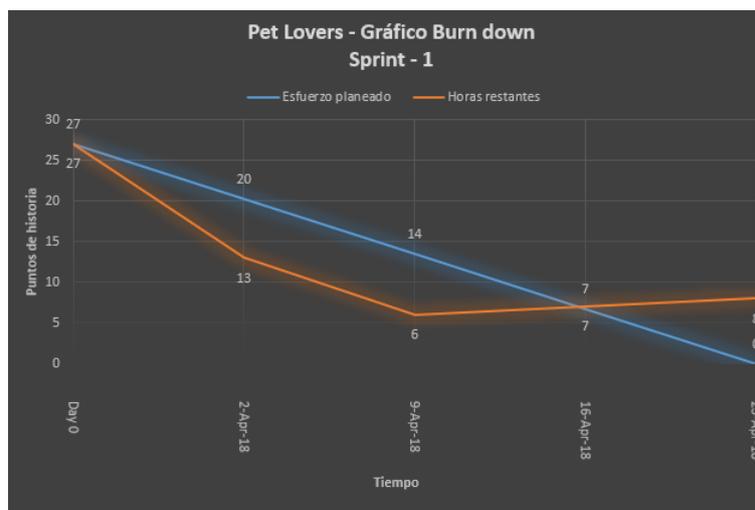


Figura 4. Gráfico de final de Burn down del Sprint 1

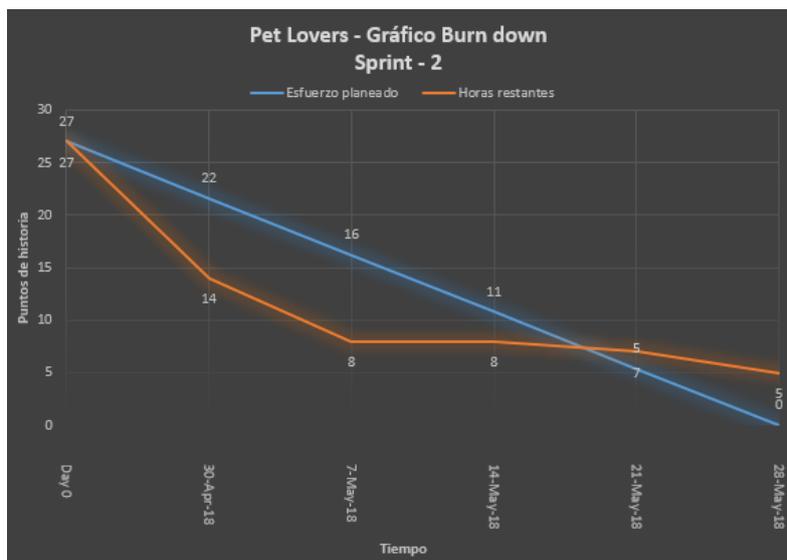


Figura 5. Gráfico de final de Burn down del Sprint 2

En el gráfico burndown se tiene que el avance ideal de un sprint estaría representado por la diagonal que reduce el esfuerzo pendiente de manera continua y progresiva hasta completar la semana del sprint.

La figura 4 y figura 5 muestra el estado final del registro semanal que se realizó para poder medir el trabajo restante a lo largo del proyecto. Se puede ver que al comienzo de ambos el esfuerzo fue superior al estimado para luego ser inferior al mismo.

2.5 Diagrama Entidad Relación

En la figura 10 se muestra el diagrama entidad-relación que se desarrolló con el fin de tener una base lógica para apoyar la aplicación. Su lectura y explicación sería la siguiente a continuación:

Un usuario posee muchas mascotas. Un usuario tiene los atributos de id único, nombre, *email* y *password*. Una mascota tiene los atributos de id único, tipo, nombre, sexo, edad, foto.

Muchos usuarios realizan muchas interacciones. Se entiende por interacción a la acción de poner gusto o disgusto a la mascota de otro usuario. Es por eso por

lo que las interacciones tienen como atributos tanto el iniciador de tal acción, como la mascota en la que se basó y el estado de dicha acción.

Muchas interacciones crean muchos *match*. Si la interacción es correspondida, es decir, si el Usuario A con la mascota A le gusta la mascota B del usuario B y viceversa, se crea un match. Los atributos por lo tanto serán el iniciador (Usuario A) de la primera interacción y el aceptador (Usuario B) de la segunda interacción.

Un *match* habilita un canal de mensajería el cual tendrá el atributo de id único, tiempo que indica la fecha del mensaje y el mensaje como tal.

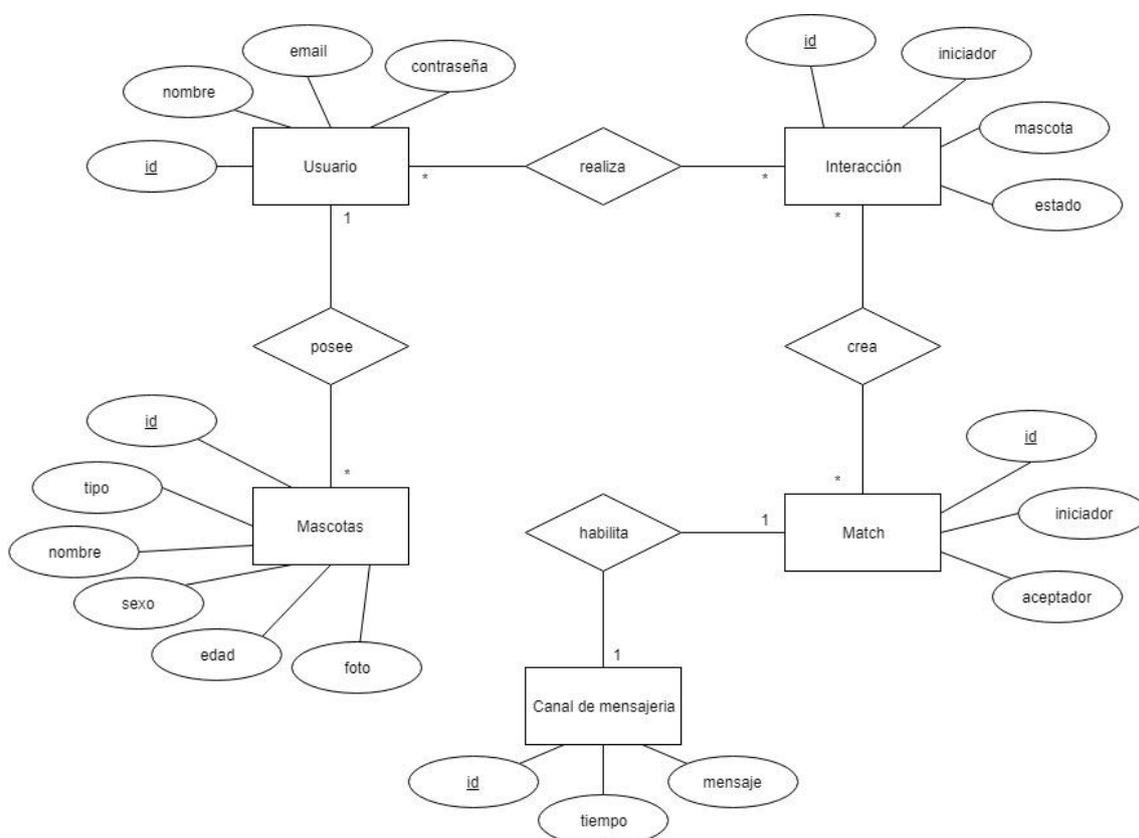


Figura 6. Diagrama entidad relación de la solución

2.6 Modelo Vista Controlador

De acuerdo con el patrón MVC, como se puede apreciar en la figura 6, se encuentra la carpeta denominada “*Providers*”, la cual consta de las clases o modelos. En la figura 7, se muestra en la carpeta “*views*” las vistas y en la figura

8, en la carpeta denominada “Controllers” existen los controladores de la solución.

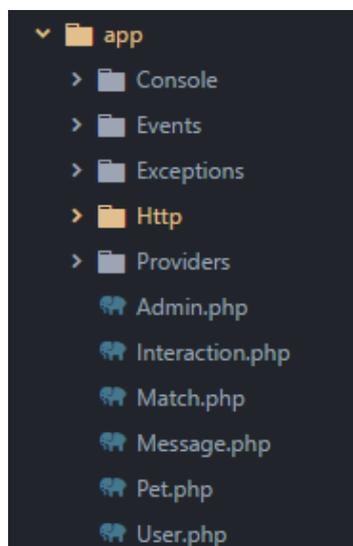


Figura 7. Modelo de la solución

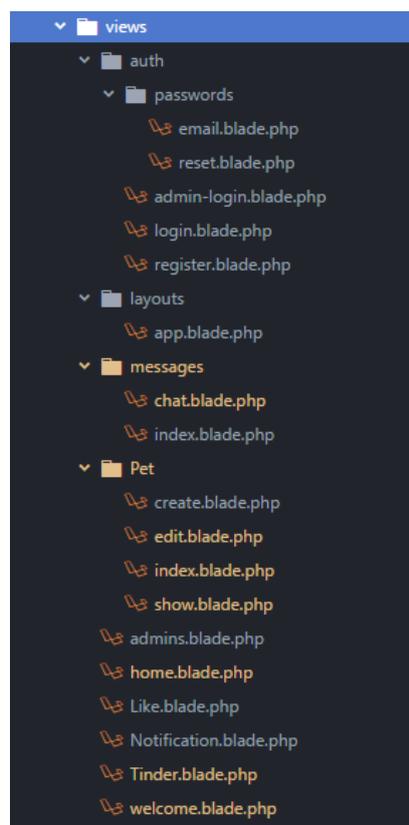


Figura 8. Vista de la solución

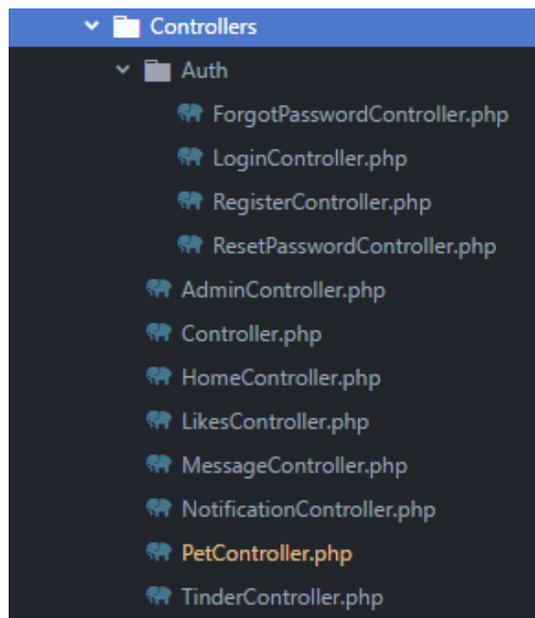


Figura 9. Controlador de la solución

En la figura 9 se muestra de manera gráfica cómo los elementos se comunican para formar el patrón MVC en Laravel. Las rutas se comunican con los controladores, estos a su vez se comunican tanto con la vista como con los modelos. La vista se muestra en las interfaces y los modelos están unidos con la base de datos.

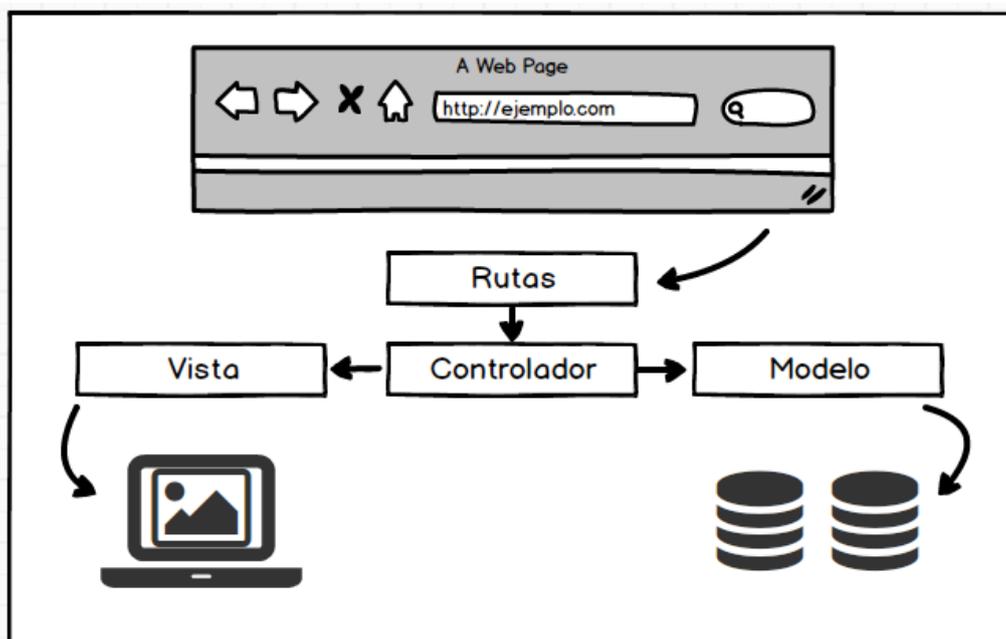
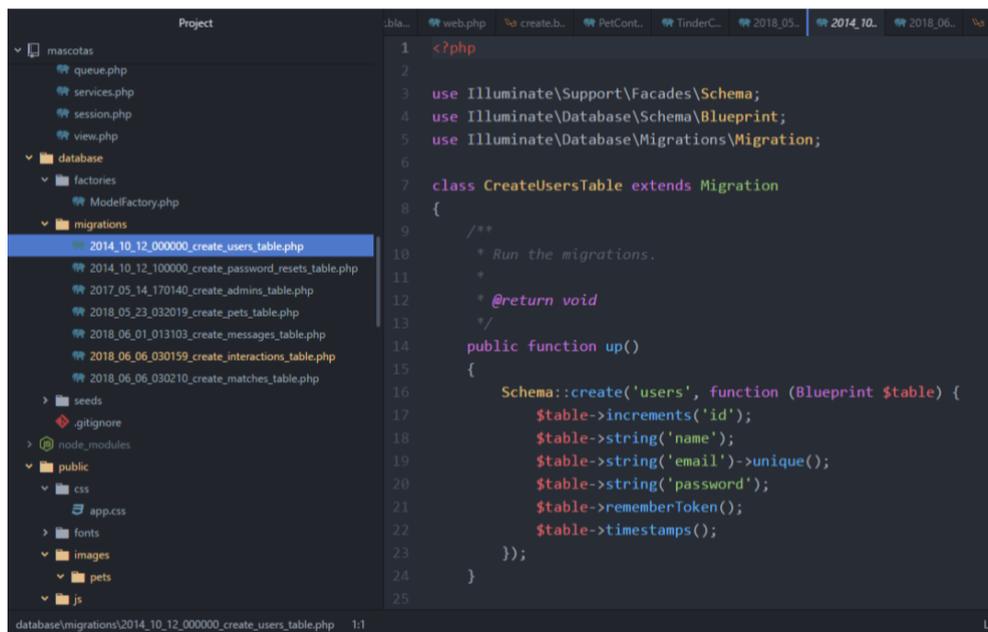


Figura 10. MVC Laravel de la solución

2.7 Desarrollo

2.7.1 Migraciones y Base de Datos

A continuación, se expondrán capturas de pantalla de los códigos para la realización de las migraciones de Laravel hacia la base de datos MySQL como se observa en la figura 11,12,13,14 y 15.

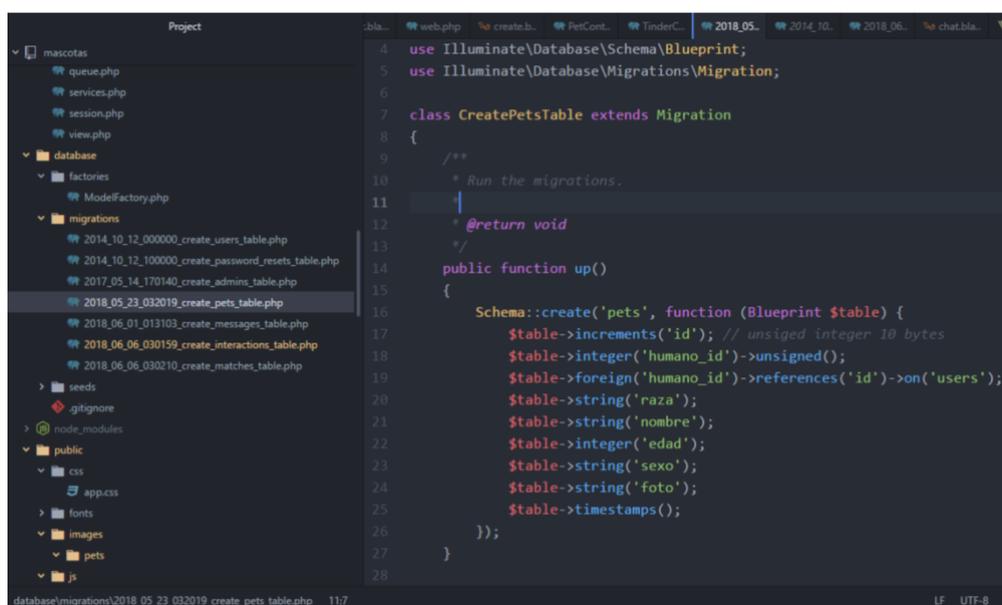


```

1  <?php
2
3  use Illuminate\Support\Facades\Schema;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Database\Migrations\Migration;
6
7  class CreateUsersTable extends Migration
8  {
9      /**
10     * Run the migrations.
11     *
12     * @return void
13     */
14     public function up()
15     {
16         Schema::create('users', function (Blueprint $table) {
17             $table->increments('id');
18             $table->string('name');
19             $table->string('email')->unique();
20             $table->string('password');
21             $table->rememberToken();
22             $table->timestamps();
23         });
24     }
25

```

Figura 11. Parte del código para el desarrollo de la tabla de usuarios.



```

4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Database\Migrations\Migration;
6
7  class CreatePetsTable extends Migration
8  {
9      /**
10     * Run the migrations.
11     *
12     * @return void
13     */
14     public function up()
15     {
16         Schema::create('pets', function (Blueprint $table) {
17             $table->increments('id'); // unsigned integer 10 bytes
18             $table->integer('humano_id')->unsigned();
19             $table->foreign('humano_id')->references('id')->on('users');
20             $table->string('raza');
21             $table->string('nombre');
22             $table->integer('edad');
23             $table->string('sexo');
24             $table->string('foto');
25             $table->timestamps();
26         });
27     }
28

```

Figura 12. Parte del código para el desarrollo de la tabla de mascotas.

```

6
7 class CreateInteractionsTable extends Migration
8 {
9     /**
10      * Run the migrations.
11      *
12      * @return void
13      */
14     public function up()
15     {
16         Schema::create('interactions', function (Blueprint $table) {
17             $table->increments('id');
18             $table->unsignedInteger('user_id');
19             $table->foreign('user_id')->references('id')->on('users');
20             $table->unsignedInteger('pet_id');
21             $table->foreign('pet_id')->references('id')->on('pets')->onDelete('CASCADE');
22             $table->boolean('is_like');
23             $table->timestamps();
24         });
25     }
26
27     /**
28      * Reverse the migrations.
29      *
30      * @return void

```

Figura 13. Parte del código para el desarrollo de la tabla de interacciones.

```

1 <?php
2
3 use Illuminate\Support\Facades\Schema;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Database\Migrations\Migration;
6
7 class CreateMatchesTable extends Migration
8 {
9     /**
10      * Run the migrations.
11      *
12      * @return void
13      */
14     public function up()
15     {
16         Schema::create('matches', function (Blueprint $table) {
17             $table->increments('id');
18             $table->unsignedInteger('user_id');
19             $table->foreign('user_id')->references('id')->on('users');
20             $table->unsignedInteger('match_user_id');
21             $table->foreign('match_user_id')->references('id')->on('users');
22             $table->timestamps();
23         });
24     }
25

```

Figura 14. Parte del código para el desarrollo de la tabla de matches.

```

1  <?php
2
3  use Illuminate\Support\Facades\Schema;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Database\Migrations\Migration;
6
7  class CreateMessagesTable extends Migration
8  {
9      /**
10     * Run the migrations.
11     *
12     * @return void
13     */
14     public function up()
15     {
16         Schema::create('messages', function (Blueprint $table) {
17             $table->increments('id');
18             $table->text('message');
19             $table->integer('match_id')->unsigned();
20             $table->integer('user_id')->unsigned();
21             $table->foreign('user_id')->references('id')->on('users');
22             $table->timestamps();
23         });
24     }
25

```

Figura 15. Parte del código para el desarrollo de la tabla de mensajes.

Como se muestra en la figura 16, se migraron las tablas a la base de datos MySQL. Se observan también algunas migraciones no nombradas anteriormente, estas son las migraciones para administradores y para el reseteo de contraseñas, pero al no ser parte de las historias de usuario sino más bien como parte de control de la solución no fueron expuestas en el documento.

```

C:\Users\v_a_1\Docs\Laravel\mascotas (master -> origin)
λ php artisan migrate
Migration table created successfully.
Migrating: 2014_10_12_000000_create_users_table
Migrated: 2014_10_12_000000_create_users_table
Migrating: 2014_10_12_100000_create_password_resets_table
Migrated: 2014_10_12_100000_create_password_resets_table
Migrating: 2017_05_14_170140_create_admins_table
Migrated: 2017_05_14_170140_create_admins_table
Migrating: 2018_05_23_032019_create_pets_table
Migrated: 2018_05_23_032019_create_pets_table
Migrating: 2018_06_01_013103_create_messages_table
Migrated: 2018_06_01_013103_create_messages_table
Migrating: 2018_06_06_030159_create_interactions_table
Migrated: 2018_06_06_030159_create_interactions_table
Migrating: 2018_06_06_030210_create_matches_table
Migrated: 2018_06_06_030210_create_matches_table

```

Figura 16. Ejecución del comando “*php artisan migrate*” para la migración.


```

10     use Notifiable;
11
12     /**
13      * The attributes that are mass assignable.
14      * @var array
15      */
16     protected $fillable = [
17         'name', 'email', 'password',
18     ];
19
20     /**
21      * The attributes that should be hidden for arrays.
22      * @var array
23      */
24     protected $hidden = [
25         'password', 'remember_token',
26     ];
27
28     public function pets(){
29         return $this->hasMany(Pet::class,'humano_id');
30     }
31
32     public function message(){
33         return $this->hasMany(Message::class);
34     }

```

Figura 18. Parte del código para el desarrollo de la clase usuario.

```

1 <?php
2
3 namespace App;
4
5 use Illuminate\Database\Eloquent\Model;
6
7 class Pet extends Model
8 {
9     protected $table = 'pets';
10    public $timestamps = false;
11
12    public function Userid(){
13        return $this->belongsTo(User::class,'humano_id');
14    }
15
16    public function owner(){
17        return $this->belongsTo(User::class,'humano_id');
18    }
19
20    public function interactions(){
21        return $this->hasMany(Interaction::class);
22    }
23
24 }
25

```

Figura 19. Parte del código para el desarrollo de la clase mascota.

The screenshot shows an IDE window with a project named 'mascotas'. The left sidebar displays a file tree with folders like 'app', 'bootstrap', 'cache', and 'config'. The 'app' folder is expanded, and 'Interaction.php' is selected. The main editor displays the following PHP code:

```

1  <?php
2
3  namespace App;
4
5  use Illuminate\Database\Eloquent\Model;
6
7  class Interaction extends Model
8  {
9      public function user(){
10         return $this->belongsTo(User::class);
11     }
12     public function pet(){
13         return $this->belongsTo(Pet::class);
14     }
15 }
16

```

Figura 20. Parte del código para el desarrollo de la clase interacción.

The screenshot shows the same IDE window, but now 'Match.php' is selected in the file tree. The main editor displays the following PHP code:

```

1  <?php
2
3  namespace App;
4
5  use Illuminate\Database\Eloquent\Model;
6
7  class Match extends Model
8  {
9      public function user(){
10         return $this->belongsTo(User::class);
11     }
12     public function match(){
13         return $this->belongsTo(User::class, 'match_user_id');
14     }
15 }
16

```

Figura 21. Parte del código para el desarrollo de la clase match.

```

1  <?php
2
3  namespace App;
4
5  use Illuminate\Database\Eloquent\Model;
6
7  class Message extends Model
8  {
9      protected $fillable = ['message', 'match_id', 'user_id'];
10
11     public function user(){
12         return $this->belongsTo(User::class);
13     }
14 }
15

```

Figura 22. Parte del código para el desarrollo de la clase mensaje.

2.7.3 Controladores

Los controladores correspondientes se muestran a continuación en las figuras 23,24,25,26 y 27.

```

11
12 class PetController extends Controller
13 {
14     /**
15      * Display a listing of the resource.
16      *
17      * @return \Illuminate\Http\Response
18      */
19     public function index()
20     {
21         //Realiza un check si esta ingresado el usuario
22         //Coje le Id del usuario = Humano en tabla pets
23         if (Auth::check()){
24             $humanoId = Auth::user()->id;
25             $queryBuilder = Pet::where('humano_id', $humanoId);
26             $pet = $queryBuilder->get();
27             return view ('Pet.index')
28                 ->with('pet', $pet);
29         }else{
30             return view ('welcome');
31         }
32     }
33
34     /**

```

Figura 23. Parte del código para el desarrollo del controlador de mascotas.

```

38     private function interaction($id,$isLike){
39         //Id del que dio Like
40         $me = Auth::user();
41         //Encuentra a Las mascota gustada
42         $petSeleccionado = Pet::find($id);
43         //Nueva notificación con los datos
44         $interaction = new Interaction();
45         $interaction->is_like = $isLike;
46         $interaction->user_id = $me->id;
47         $interaction->pet_id = $petSeleccionado->id;
48         $interaction->save();
49
50         $owner = $petSeleccionado->owner;
51         $pets = $me->pets;
52         $myPetsIds = $pets->pluck('id');
53         $hasMatch = Interaction::where('user_id',$owner->id)
54             ->where('is_like',true)
55             ->whereIn('pet_id',$myPetsIds)->first();
56         if($hasMatch != null && $isLike){
57             $match = new Match();
58             $match->user_id = $me->id;
59             $match->match_user_id = $owner->id;
60             $match->save();

```

Figura 24. Parte del código para el desarrollo del controlador de matches.

```

13 class NotificationController extends Controller
14 {
15     //
16     public function index()
17     {
18         //Realiza un check si esta ingresado el usuario
19         if (Auth::check()){
20             //Listado de notificaciones hacia este usuario.
21             $me = Auth::user()->id;
22             $pets = Pet::where('humano_id',$me)->get();
23             $myPetsIds = $pets->pluck('id');
24             $hasLike = Interaction::where('is_like',true)
25                 ->whereIn('pet_id',$myPetsIds)->get();
26             return view ('Notification')->with('notifications', $hasLike);
27         }else{
28             return view ('welcome');
29         }
30     }
31
32     public function profile($id)
33     {
34         //Realiza un check si esta ingresado el usuario
35         if (Auth::check()){
36             //Listado de notificaciones hacia este usuario.
37             $profile = Pet::where('humano_id', $id) ->get();

```

Figura 25. Parte del código para el desarrollo del controlador de notificaciones.

```

12
13 class LikesController extends Controller
14 {
15     //
16     public function index()
17     {
18         //Realiza un check si esta ingresado el usuario
19         //Se manda la lista de notificaciones
20         if (Auth::check()){
21             //Listado de notificaciones hacia este usuario.
22             $me = Auth::user()->id;
23             $like = Interaction::where('user_id', $me)->get();
24             //Mirar a las notificaciones
25             return view ('Like')->with('like', $like);
26         }else{
27             return view ('welcome');
28         }

```

Figura 26. Parte del código para el desarrollo del controlador de historial.

```

12 {
13
14     public function index(){
15         $me = Auth::user();
16         $matches = Match::where('user_id', $me->id)->orWhere('match_user_id',$me->id)->get();
17         return view('messages.index')->with(['matches' => $matches]);
18     }
19     public function chat($id){
20         return view('messages.chat')->with(['chat' => $id]);
21     }
22     public function fetch($id){
23         return Message::where('match_id',$id)->with('user')->get();
24     }
25
26     public function sendMessage(Request $request,$id){
27         $user = Auth::user();
28
29         $message = Message::create([
30             'message' => $request->message,
31             'match_id' => $id,
32             'user_id' => Auth::user()->id
33         ]);
34
35         broadcast(new MessageSendEvent($user,$message))->toOthers();
36     }

```

Figura 27. Parte del código para el desarrollo del controlador de mensajes.

2.7.4 Rutas

La mayoría de las rutas expuestas se encuentran en la carpeta de *routes*, en el archivo *web.php* como se muestra en la figura 28.

```

23 Route::middleware(['auth'])->group(function () {
24
25     //Pagina de Likes
26     Route::get('Like', 'LikesController@index');
27
28     //Pagina Notificaciones
29     Route::get('Notification', 'NotificationController@index');
30
31     //Boton de Like
32     Route::get('Tinder/like/{id}', 'TinderController@like');
33
34     //Boton de Dislike
35     Route::get('Tinder/dislike/{id}', 'TinderController@dislike');
36
37     //Inicio de Tinder
38     Route::get('Tinder', 'TinderController@index')->name('patitas.i
39
40     //TODAS LAS RUTAS DE CRUD MASCOTAS
41     Route::resource('Pet', 'PetController');
42
43     //rutas chat
44     Route::get('/messages', 'MessageController@index')->name('messag

```

Figura 28. Parte del código para el desarrollo de las rutas de la solución.

2.7.5 Vistas

Las vistas de la solución se muestran en las figuras 29,30,31,32,33,34 y 35.

```

63 <div class="flex-center position-ref full-height">
64     @if (Route::has('login'))
65         <div class="top-right links">
66             @if (Auth::check())
67                 <a href="{{ url('/home') }}">Home</a>
68
69             @else
70                 <a href="{{ url('/login') }}">Ingresar</a>
71                 <a href="{{ url('/register') }}">Registrarse</a>
72             @endif
73         </div>
74     @endif
75
76     <div><div>
77
78         <div class="content">
79             <h1>
80                 Pet Lovers
81             </h1>
82             <center>
83                 
23     <h3>
24     Notificaciones
25     </h3>
26     <p>
27     En esta opción podras ver que usuarios le han dado "like" e
28     </p>
29     </div>
30     </div>
31     </div>

```

Figura 30. Parte del código para el desarrollo de la vista de inicio.

```

1 @extends('layouts.app')
2 @section('content')
3 <h1>Ingresar mascota</h1>
4 {{ Html::ul($errors->all()) }}
5 {{ Form::open(array('url' => 'Pet', 'files' => true)) }}
6 <div class="form-group">
7     {{ Form::label('raza', 'Tipo') }}
8     {{ Form::select('raza', array('Desconocido' => 'Seleccione el tipo',
9     'Perro' => 'Perro',
10    'Gato' => 'Gato',
11    'Peces' => 'Peces',
12    'Pequeños mamiferos' => 'Pequeños mamiferos',
13    'Aves' => 'Aves',
14    'Reptiles y anfibios' => 'Reptiles y anfibios'
15    ), Input::old('raza'), array('class' => 'form-control')) }}
16 </div>
17
18 <div class="form-group">
19     {{ Form::label('nombre', 'Nombre') }}
20     {{ Form::text('nombre', Input::old('nombre'), array('class' => 'form-
21 </div>
22
23 <div class="form-group">
24     {{ Form::label('edad', 'Edad') }}
25     {{ Form::text('edad', Input::old('edad'), array('class' => 'form-cont

```

Figura 31. Parte del código para el desarrollo de la vista de creación de mascota.

```

Project
├── cotas
│   ├── passwords
│   │   ├── email.blade.php
│   │   ├── reset.blade.php
│   │   ├── admin-login.blade.php
│   │   ├── login.blade.php
│   │   └── register.blade.php
│   ├── layouts
│   ├── messages
│   └── Pet
│       ├── create.blade.php
│       ├── edit.blade.php
│       ├── index.blade.php
│       ├── show.blade.php
│       ├── admins.blade.php
│       ├── home.blade.php
│       ├── Like.blade.php
│       ├── Notification.blade.php
│       └── Tinder.blade.php
├── routes
│   ├── api.php
│   ├── channels.php
│   ├── console.php
│   ├── web.php
│   └── storage
└── views
    └── Tinder.blade.php

```

```

26 <div>
27 @if(count($pet)==0)
28 <center><h2>Pronto existirán más mascotas para darles patita!</h2></center>
29 @else
30 </div>
31 <div class="row">
32 @foreach($pet as $key => $value)
33 <div class="col-md-4">
34 <div class="card" >
35 
36 <div class="card-body" style="text-align:center">
37 <h5 class="card-title">{{ $value->nombre }} ({{ $value->edad }} años)</h5>
38 <p class="card-text">{{ $value->raza }}, {{ $value->sexo }}</p>
39 <a class="btn btn-small btn-success" href="{{ URL::to('Tinder/like/' . $value->id) }}">
40 <i class="fas fa-paw"></i>
41 </a>
42 <a class="btn btn-dark" href="{{ URL::to('Tinder/dislike/' . $value->id) }}">
43 <i class="fas fa-thumbs-down"></i>
44 </a>
45 </div>
46 </div>
47 </div>
48 @endforeach
49 </div>
50

```

Figura 32. Parte del código para el desarrollo de la vista de interacciones y matches.

```

Tinder.... | Notific... | Notific... | Like.bla... | app.bla... | LikesCo... | web.php | create... | Pet
1 @extends('layouts.app')
2 @section('content')
3 @if(count($notifications)==0)
4 <center><h2>No tiene notificaciones aún</h2></center>
5 @else
6 <h1>Notificaciones</h1>
7 <hr>
8 @foreach($notifications as $notification)
9 <div class="alert alert-primary" role="alert">
10 El usuario
11 <a href="{{ URL::to('Notification/profile/' . $notification->user->id) }}">
12 {{ $notification->user->name}}
13 </a>
14 le dió patita a tu mascota {{ $notification->pet->nombre}}.
15 </div>
16
17 @endforeach
18 @endif

```

Figura 33. Parte del código para el desarrollo de la vista de notificaciones.

```

4 <h2><center>No tiene historial aún<center></h2>
5 @else
6 <h1>Historial</h1>
7 <hr>
8 @foreach($like as $like)
9     <div class="alert alert-primary" role="alert">
10         <a href="#" class="close" data-dismiss="alert" aria-label="close">&times;</a>
11         Has dado
12         @if(($like->is_like)==1)
13             patita
14         @else
15             dislike
16         @endif
17         a {{$like->pet->nombre}}.
18         Su dueño es el usuario {{ $like->pet->owner->name}}
19         como dueño. Esta acción fue realizada el día {{ $like->created_at }}.
20     </div>
21 @endforeach
22 @endif
23 @endsection

```

Figura 34. Parte del código para el desarrollo de la vista de historial.

```

5 <h1 class="text-center">Mensajes</h1>
6
7 <table class="table">
8     <thead>
9         <tr>
10            <th>Nombre</th>
11            <th><h5>Fecha</h5></th>
12        </tr>
13    </thead>
14    <tbody>
15        @foreach($matches as $match)
16            <tr>
17                @if($match->user_id == Auth::user()->id)
18                    <td><a href="{{route('messages.chat',$match->id)}}">
19                @else
20                    <td><a href="{{route('messages.chat',$match->id)}}">
21                @endif
22                    <td>{{$match->created_at}}</td>
23            </tr>
24        @endforeach
25    </tbody>
26 </table>
27
28 @endsection
29

```

Figura 35. Parte del código para el desarrollo de la vista de mensajes.

2.8 Interfaces de usuario

Las interfaces de la solución se muestran en las figuras 36,37,38,39,40,41,42 y 43,44,45,46,47 y 48.



Figura 36. Interfaz de bienvenida.

The image shows the registration form on the PetLovers website. At the top left, it says 'PetLovers' and at the top right, 'Ingresar Registrarse'. Below this is a navigation bar with links: 'Inicio', 'Mis mascotas', 'Ingresar mascota', 'Patitas', 'Notificaciones', 'Historial', and 'Mensajes'. The main content area is titled 'Registrarse' and contains the following fields: 'Nombre', 'Correo Electrónico', 'Contraseña', and 'Repetir Contraseña'. Each field has a corresponding input box. At the bottom of the form is a blue button labeled 'Registrarse'.

Figura 37. Interfaz de registro.

The image shows a login form titled "Ingreso". It contains two input fields: "Correo Electrónico" and "Contraseña". Below the password field is a checkbox labeled "Recordarme". At the bottom left is a blue button labeled "Ingresar", and to its right is a link labeled "Olvido su contraseña?".

Figura 38. Interfaz de ingreso.

The image shows the home interface of the PetLovers application. At the top left, the text "PetLovers" is displayed. At the top right, the name "Roberto" is shown with a dropdown arrow. Below this is a dark navigation bar with the following menu items: "Inicio", "Mis mascotas", "Ingresar mascota", "Patitas", "Notificaciones", "Historial", and "Mensajes". The main content area features a large heading "Bienvenido a Pet Lovers!". Below this heading are several sections, each with a title and a brief description: "Mis mascotas" (describing viewing and editing pets), "Nueva Mascota" (describing adding a new pet), "Patitas" (describing giving likes or 'patitas'), "Notificaciones" (describing seeing notifications), "Historial" (describing seeing activity), and "Mensajería" (partially visible).

Figura 39. Interfaz de inicio.

Ingresar mascota

Tipo

Seleccione el tipo ▼

Nombre

Edad

Sexo

Seleccione sexo ▼

Foto Ningún archivo seleccionado

Figura 40. Interfaz de ingreso de mascota.

PetLovers Roberto ▼

Inicio Mis mascotas Ingresar mascota Patitas Notificaciones Historial Mensajes

Todas tus mascotas



Adrian (1 años)
Perro, Masculino

Figura 41. Interfaz de mascotas.

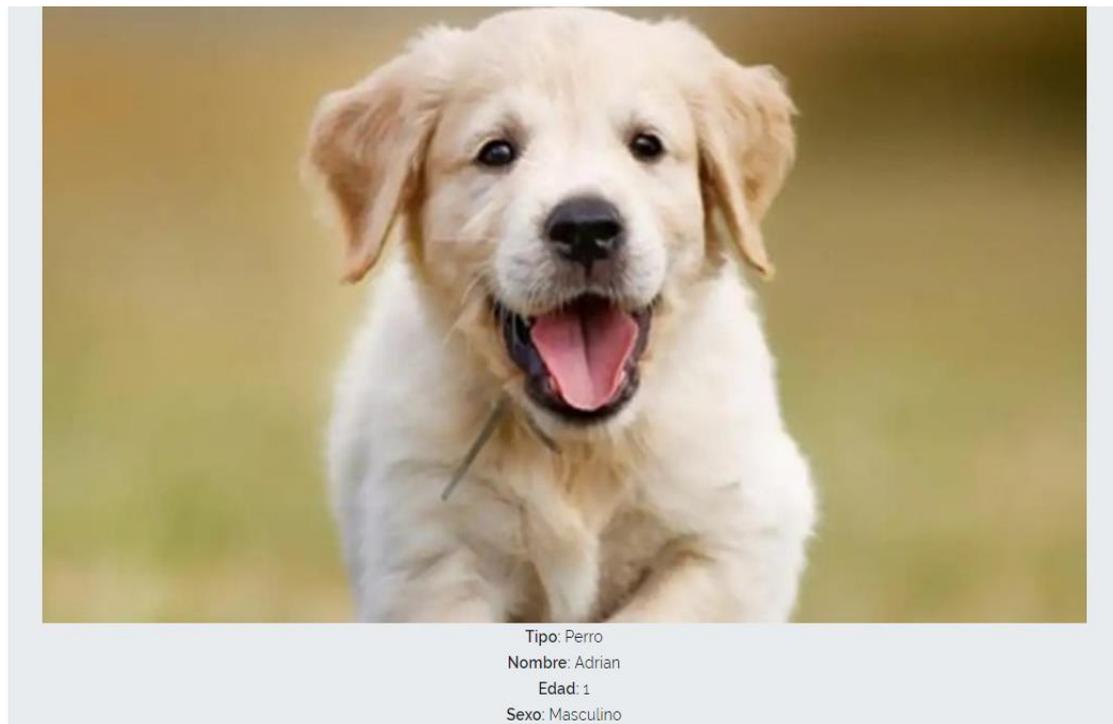


Figura 42. Interfaz de mostrar mascota.

Modificar Adrian

raza

Perro

Nombre

Adrian

Edad

1

Sexo

Masculino

Foto Ningún archivo seleccionado

Figura 43. Interfaz de editar mascota.

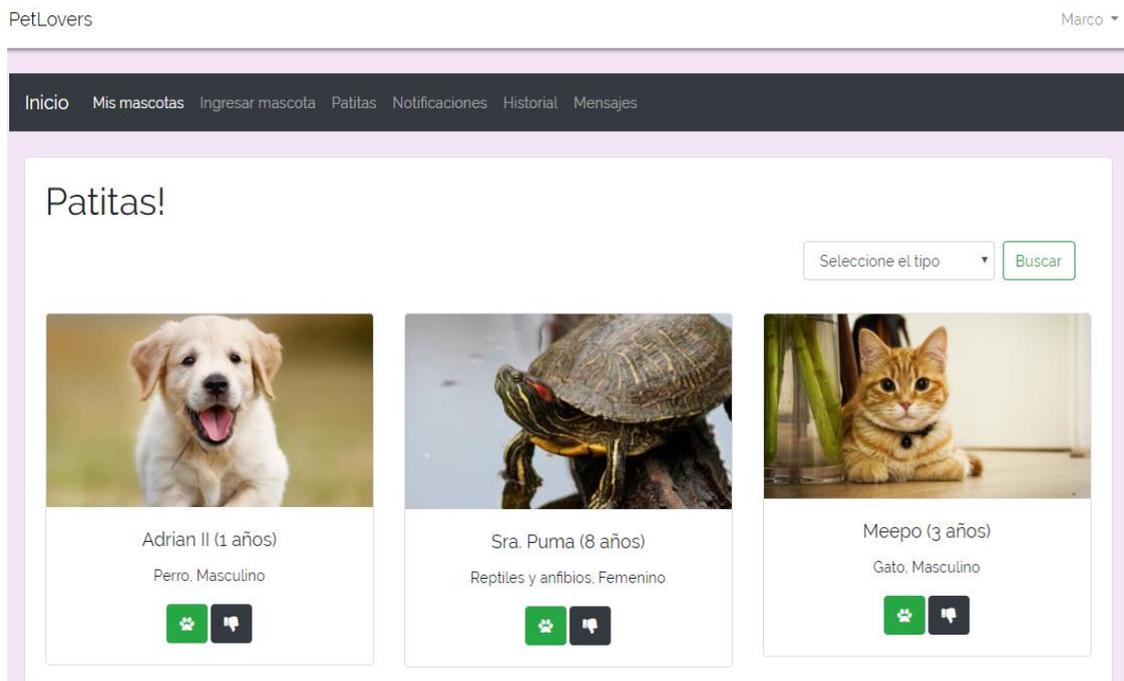


Figura 44. Interfaz de patitas para las interacciones y matches.

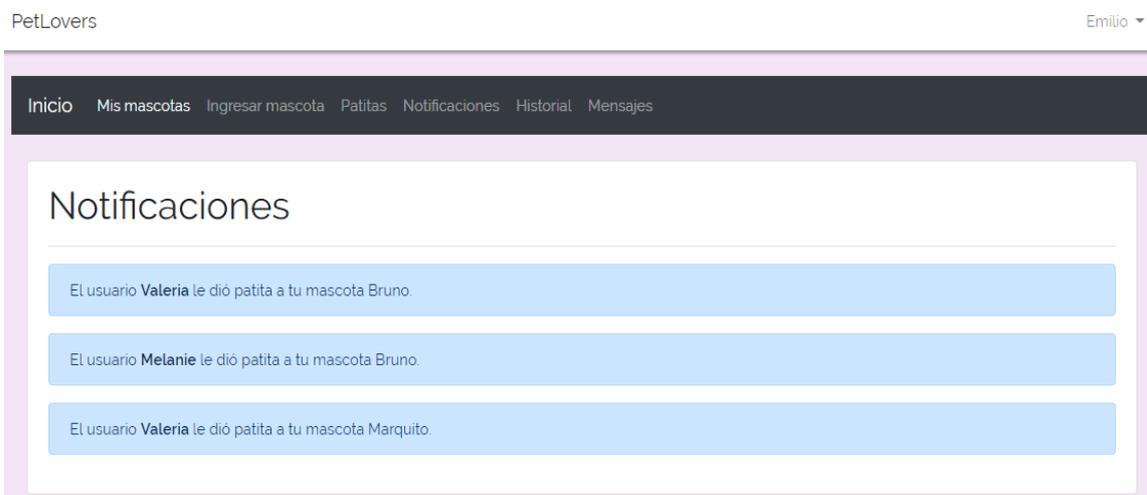


Figura 45. Interfaz de notificaciones.

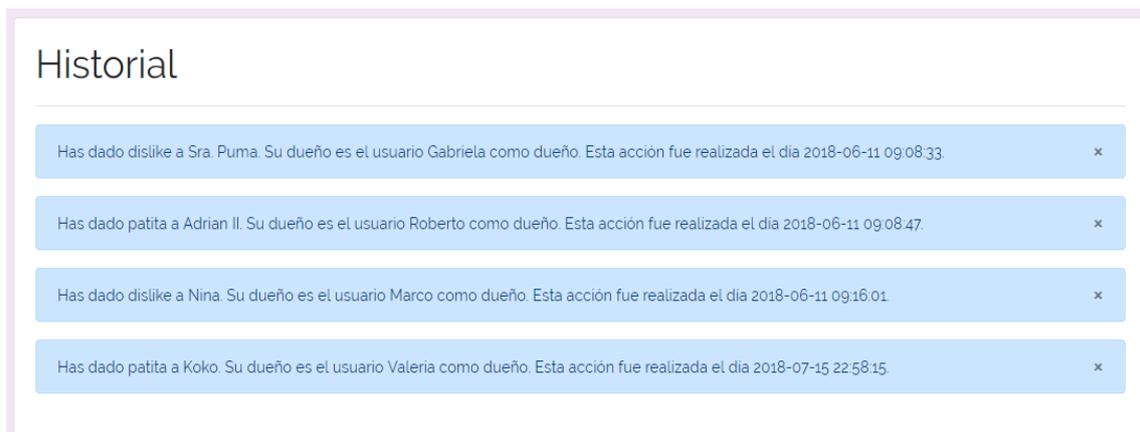


Figura 46. Interfaz de historial.

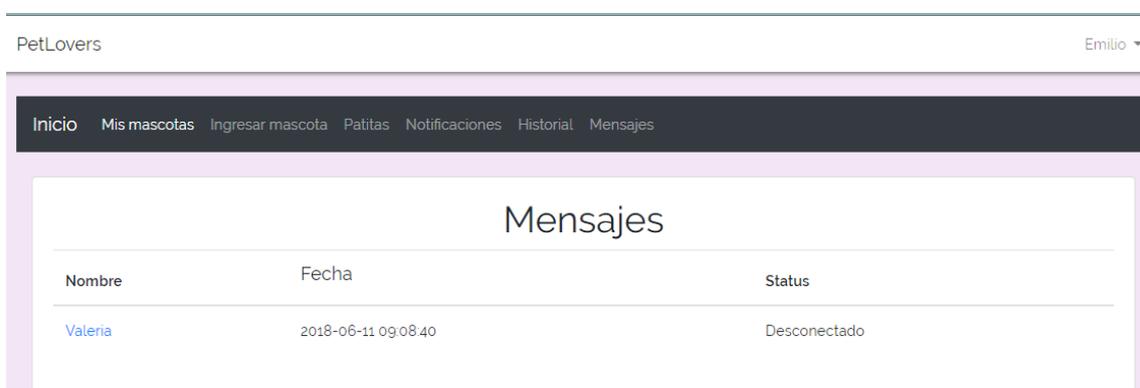


Figura 47. Interfaz de mensajes.



Figura 48. Interfaz del canal de chat.

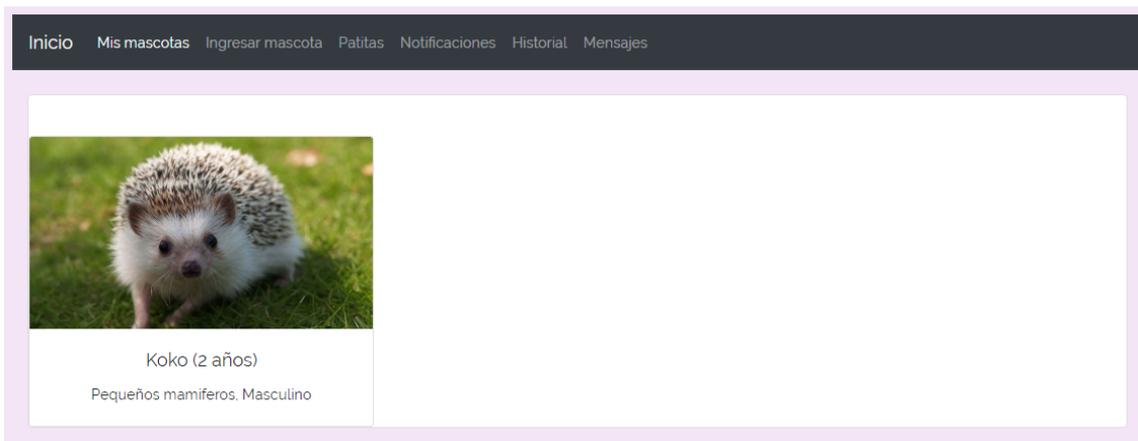


Figura 49. Interfaz de mascotas del usuario.

3.CAPÍTULO III. Validación

3.1 Introducción

En este capítulo, se procederá a validar la propuesta de solución planteada con el objetivo de validar la aplicación web de red social para estimular los vínculos afectivos entre mascotas y sus dueños.

Las pruebas serán tanto unitarias (realizadas por el programador) como pruebas de aceptación del cliente. Se documentará la realización de cada una de las pruebas y sus resultados respectivos. Se realizará un análisis por cada prueba mostrada y se describirá la prueba.

3.2 Pruebas de aceptación del cliente

A tipo de manual de usuario en la tabla 4, el siguiente listado de pasos encapsula las historias de usuario descritas en el anterior capítulo y a la vez marca una guía para la utilización de la solución. Si se realizan con éxito los pasos, en el estado se mostrará la palabra "Correcto" caso contrario "Falla".

Tabla 4.

Listado de pasos para el cliente y su estado.

Paso	Detalle	Estado
1	En la página de bienvenida hacer clic en el botón de registro, ingresar nombre, email, y la contraseña dos veces, oprimir “Registrarse” e ingresar al sistema.	Correcto
2	Cerrar sesión e ir a la opción de iniciar, ingresar el email y la contraseña establecida en el paso 1 y dar clic al botón de “Ingresar” e ingresar al sistema.	Correcto
3	Seleccionar la opción del menú “Ingresar mascota”, llenar los campos indicados y verificar su ingreso en la pestaña de “Mis mascotas”.	Correcto
4	Hacer clic en la página de “Mis mascotas”, identificar la mascota ingresada en el paso 3 y oprimir el botón con el ícono de ojo, se redireccionará al perfil de la mascota.	Correcto
5	Seleccionar la opción del menú “Mis mascotas”, identificar la mascota ingresada en el paso 3, oprimir el botón con el icono de lápiz y una vez redireccionado al formulario de edición, cambiar cualquier campo y verificar el cambio en la página de “Mis mascotas”.	Correcto
6	Ir a la opción de “patitas” del menú y filtrar por tipo de mascota seleccionando en el combo box ubicado en la parte superior derecha.	Correcto
7	Ir a la opción de “patitas” del menú y reaccionar a las mascotas de la red social dependiendo de los gustos del usuario. Verificar la actividad en la página de “Historial”.	Correcto

8	Ir a la pestaña de “Notificaciones”, esperar la reacción de “patita” por otro usuario, buscar su mascota en el sistema y reaccionar positivamente también. Verificar el mensaje de “match” por parte del sistema.	Correcto
9	Hacer clic en la opción de “Mensajes”, identificar al usuario con el que se hizo match en el paso 8, hacer clic en su nombre y mandar un mensaje.	Correcto
10	Esperar la respuesta del mensaje por parte del usuario con el que se realizó el match. Entablar una conversación y verificar que reingresando a la opción del menú la conversación siga guardada.	Correcto
11	Ir a “notificaciones” y dar clic en el enlace de un usuario cualquiera para ver sus mascotas.	Correcto

En base a la tabla 4, las imágenes siguientes capturan una demostración de los pasos realizados en la misma siendo estas una verificación del cumplimiento correcto de la solución. Cumpliendo con los requisitos o historias de usuario marcadas en el capítulo 2 del actual documento.

PetLovers Ingresar Registrarse

Inicio Mis mascotas Ingresar mascota Patitas Notificaciones Historial Mensajes

Registrarse

Nombre

Correo Electrónico

Contraseña

Repetir Contraseña

[Registrarse](#)

Figura 50. Demostración del paso 1 parte primera.

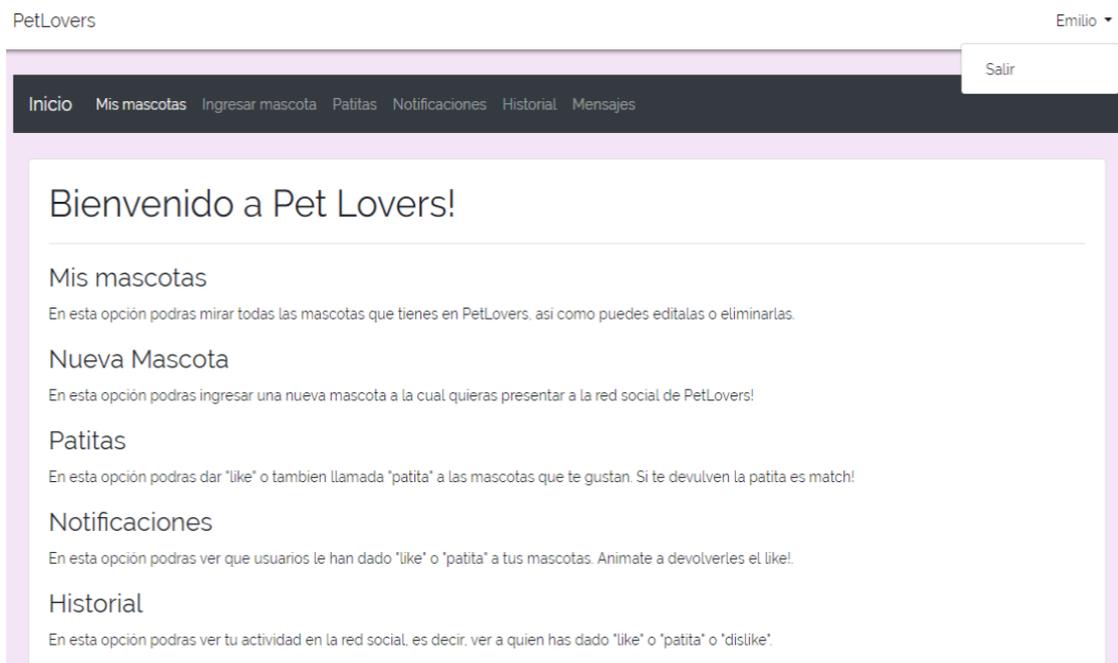


Figura 51. Demostración del paso 1 parte segunda.

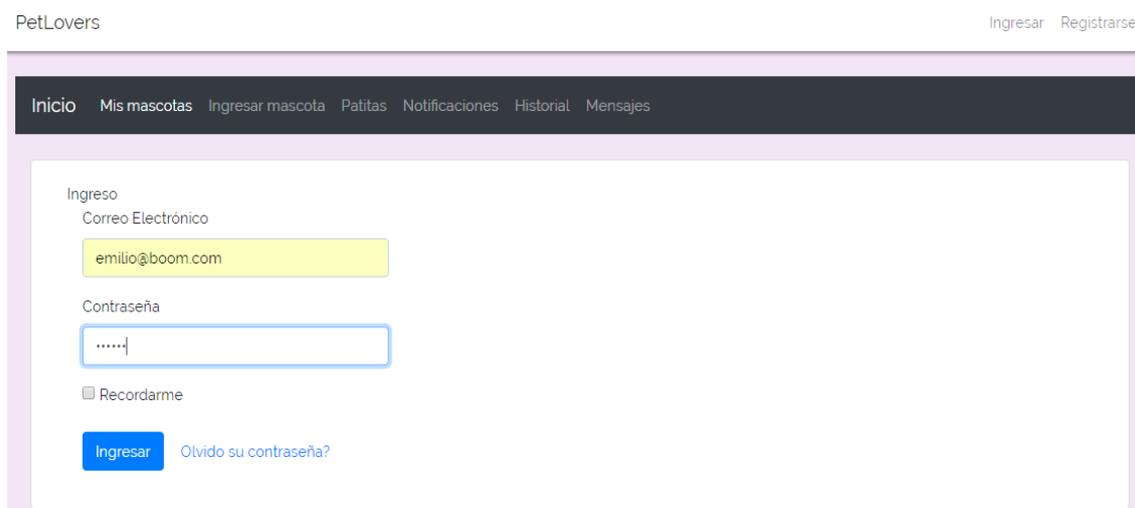


Figura 52. Demostración del paso 2 parte primera.

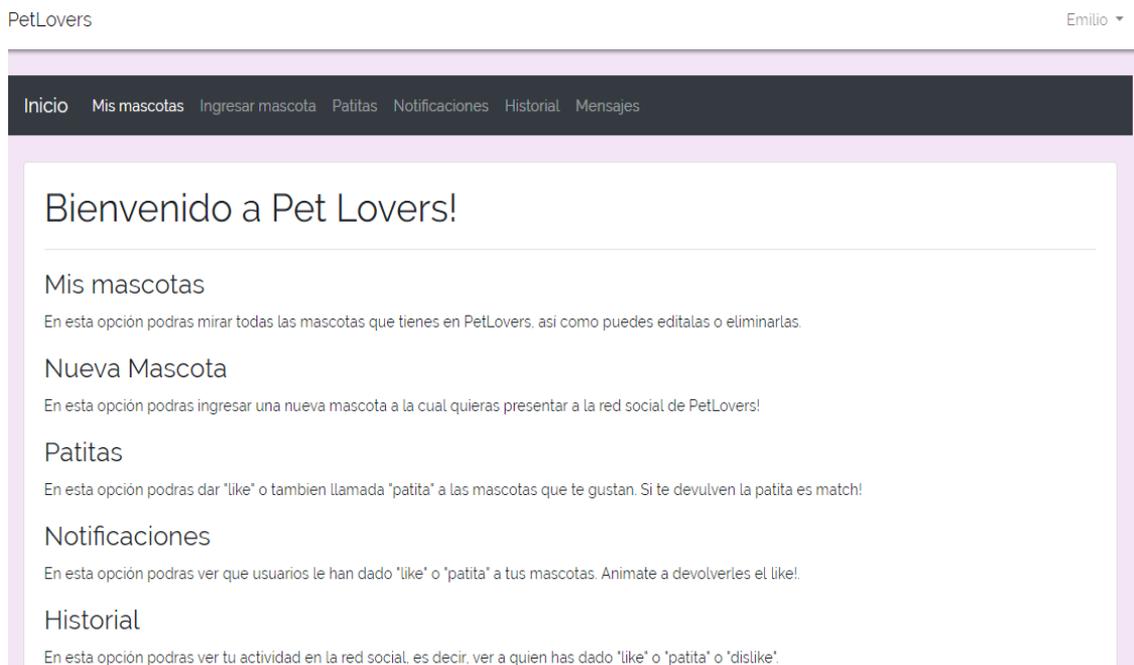


Figura 53. Demostración del paso 2 parte segunda.

The screenshot shows the "Ingresar mascota" form. It includes the following fields: "Tipo" (dropdown menu with "Pequeños mamíferos" selected), "Nombre" (text input with "Bruno"), "Edad" (text input with "5"), "Sexo" (dropdown menu with "Masculino" selected), and "Foto" (file selection button labeled "Seleccionar archivo" and a filename "descarga.jpg"). A blue "Crearlo!" button is located at the bottom of the form.

Figura 54. Demostración del paso 3 parte primera.

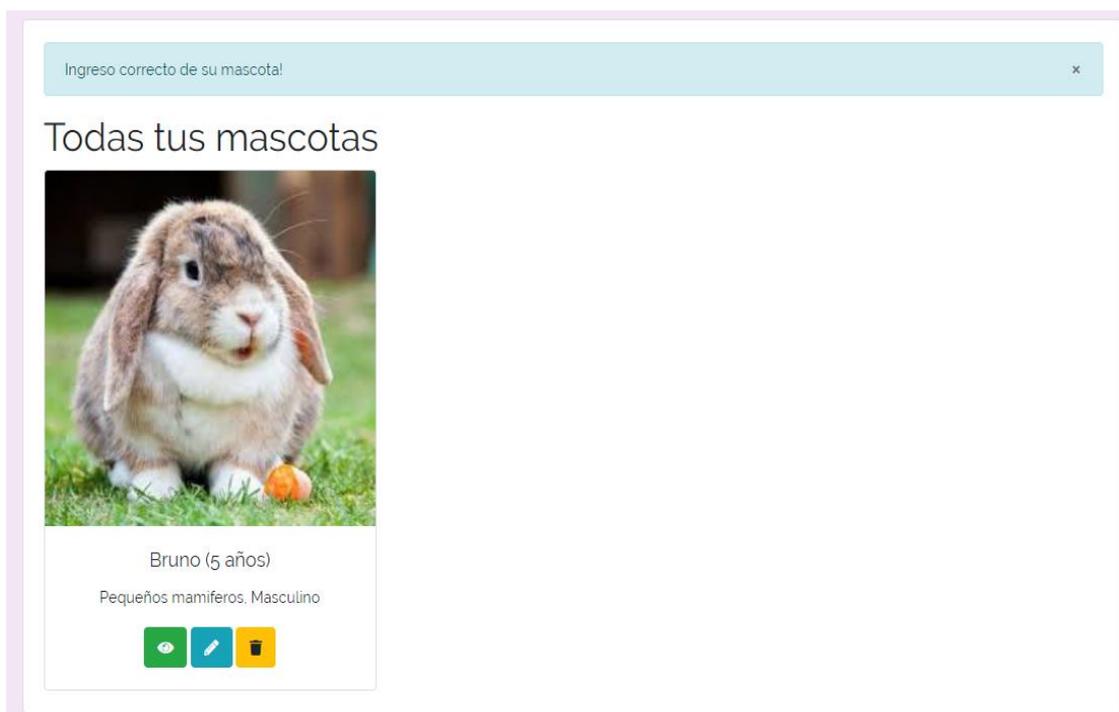


Figura 55. Demostración del paso 3 parte segunda.

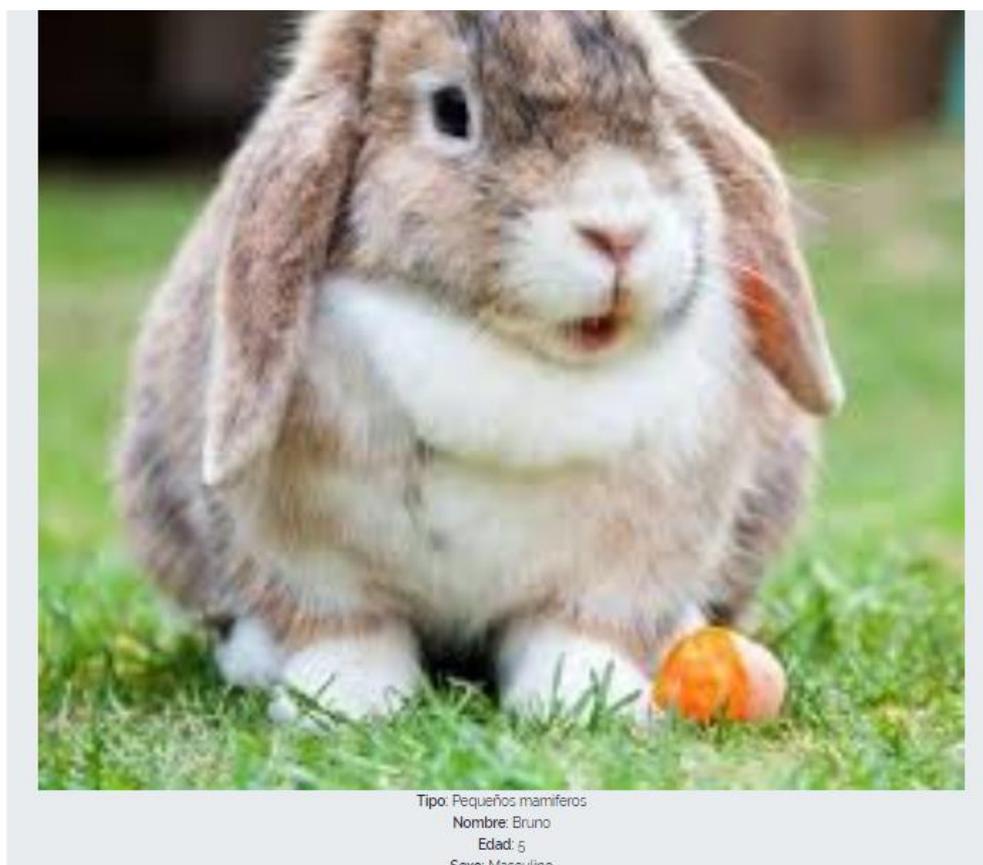


Figura 56. Demostración del paso 4.

Modificar Bruno

raza

Pequeños mamíferos

Nombre

Bruno

Edad

7

Sexo

Masculino

Foto Ningún archivo seleccionado

Figura 57. Demostración del paso 5 parte primera.

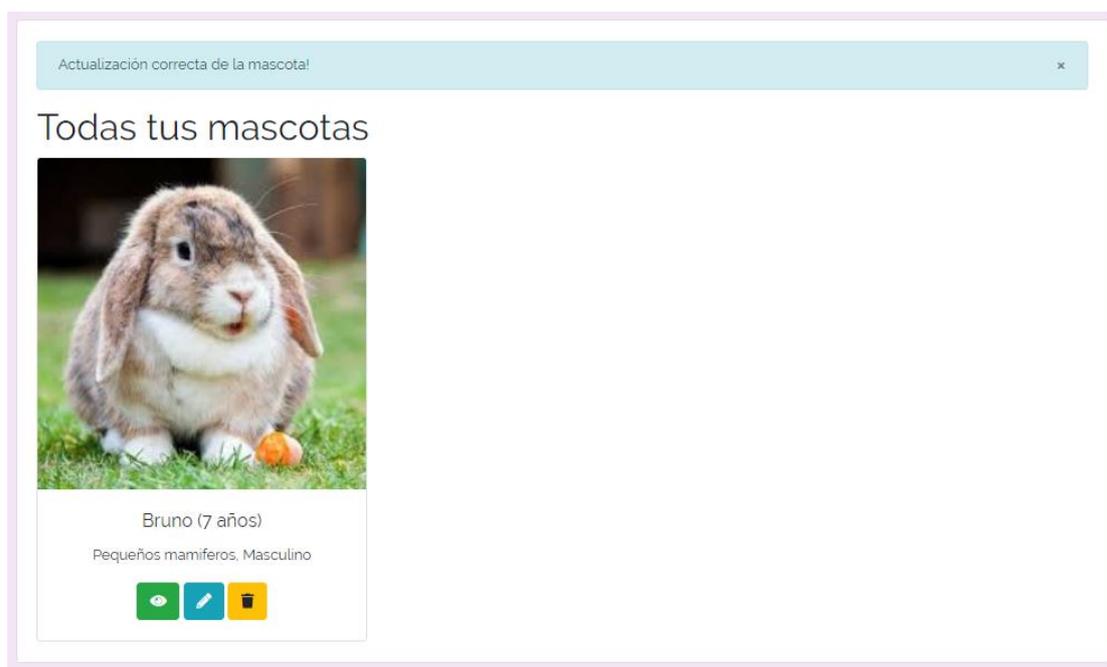


Figura 58. Demostración del paso 5 parte segunda.

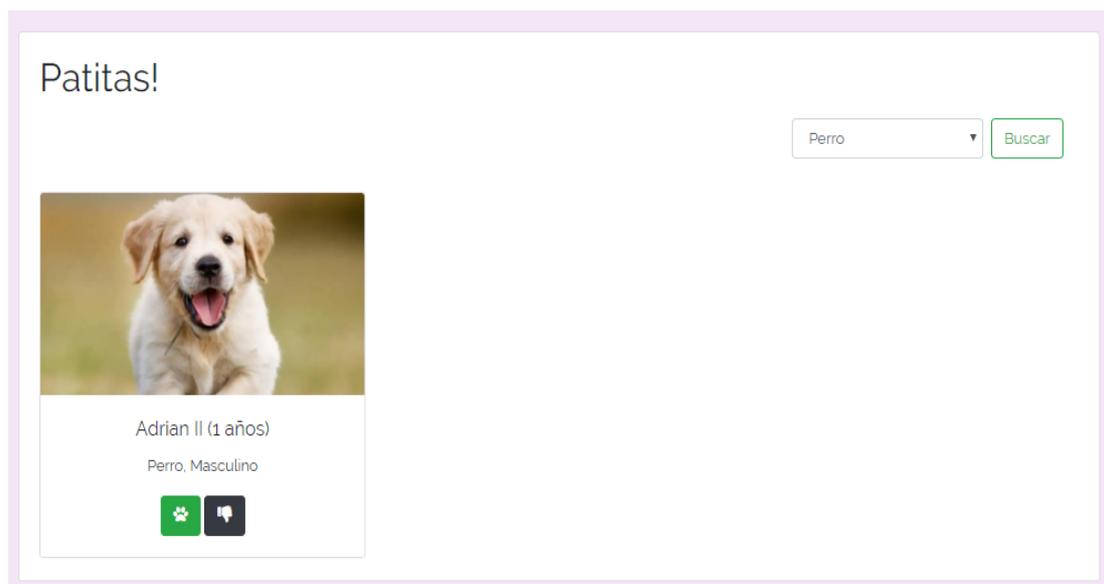


Figura 59. Demostración del paso 6 parte primera.

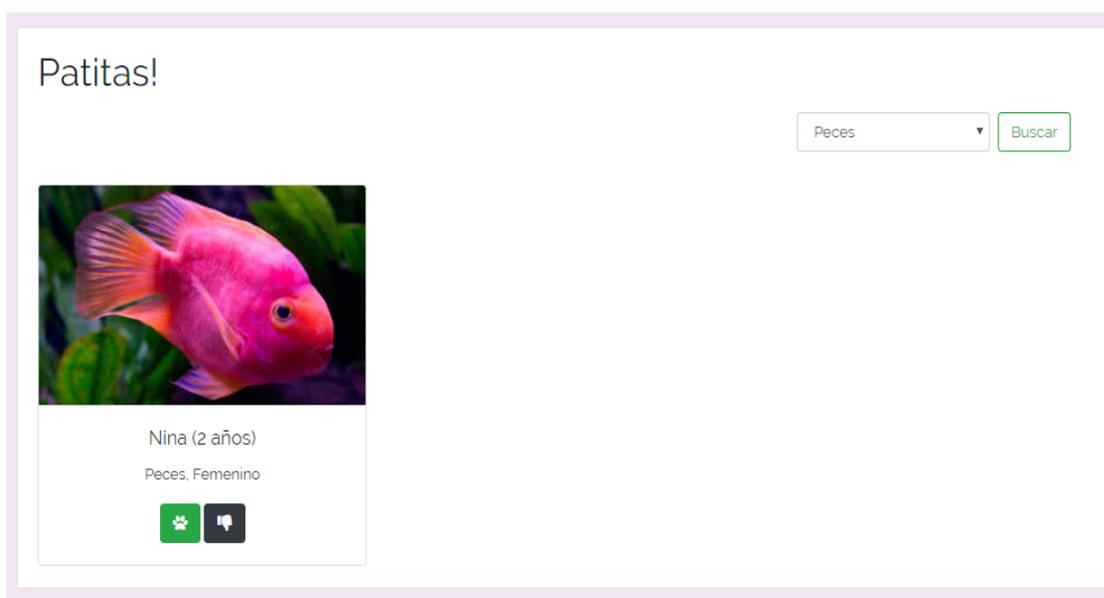


Figura 60. Demostración del paso 6 parte segunda.

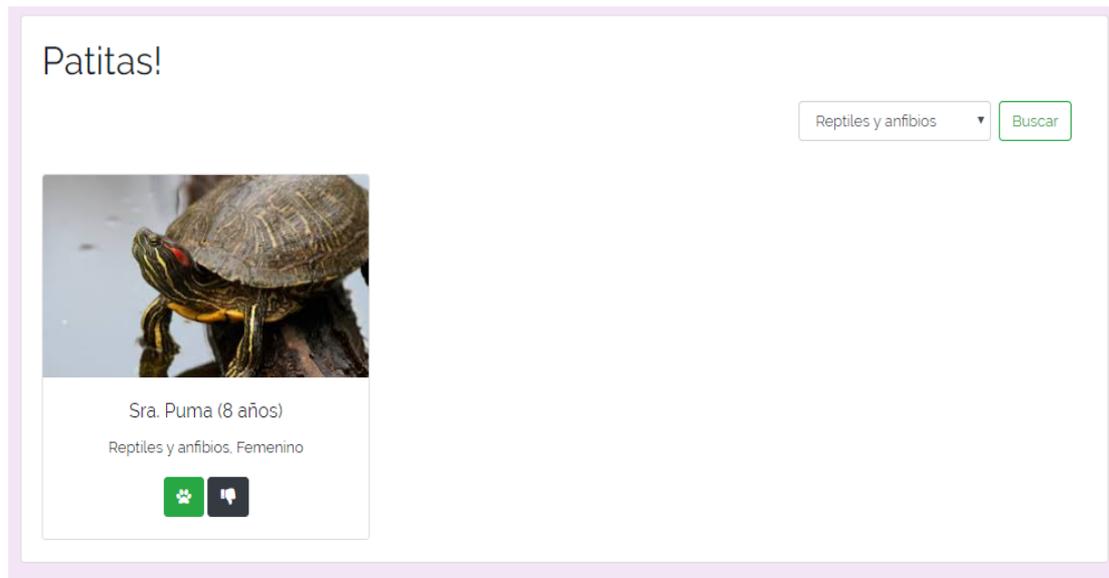


Figura 61. Demostración del paso 6 parte tercera.

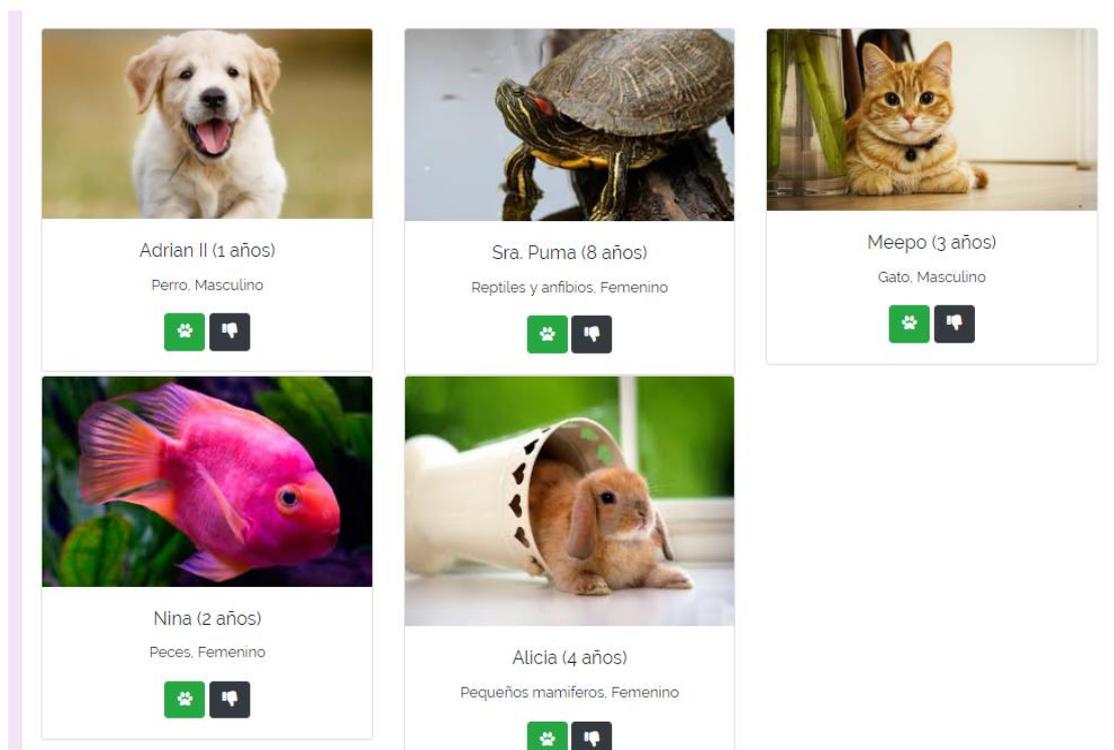


Figura 62. Demostración del paso 7 parte primera.

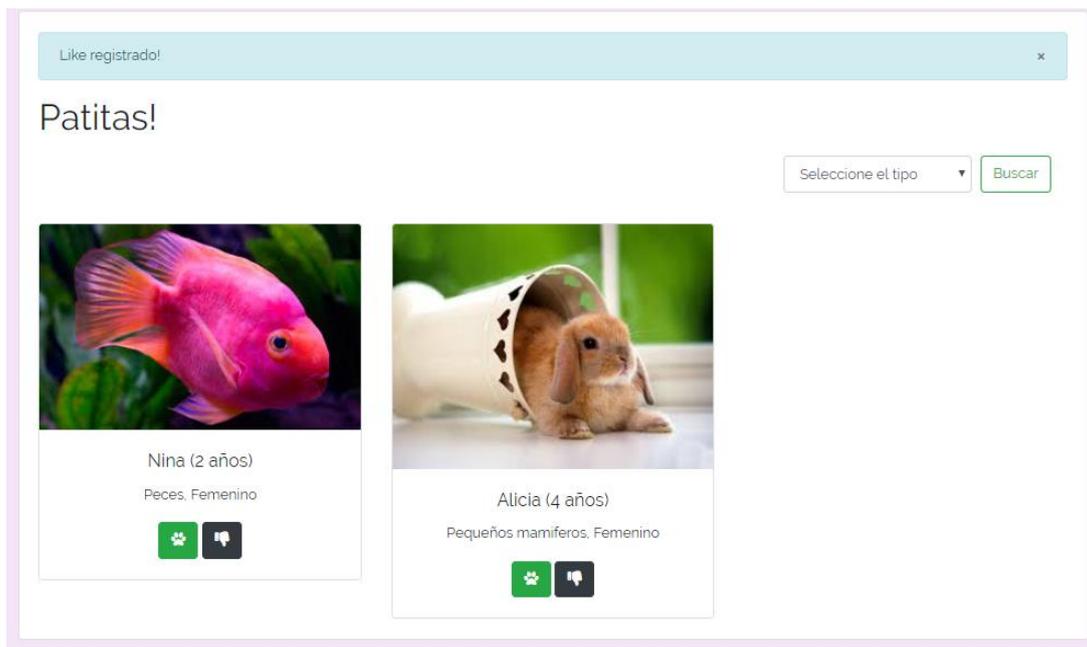


Figura 63. Demostración del paso 7 parte segunda.

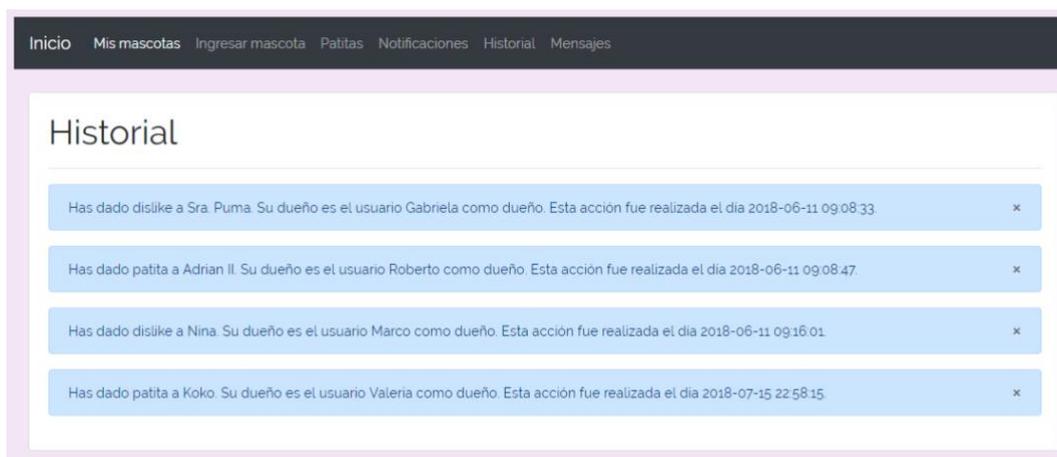


Figura 64. Demostración del paso 7 parte tercera.



Figura 65. Demostración del paso 8 parte primera.

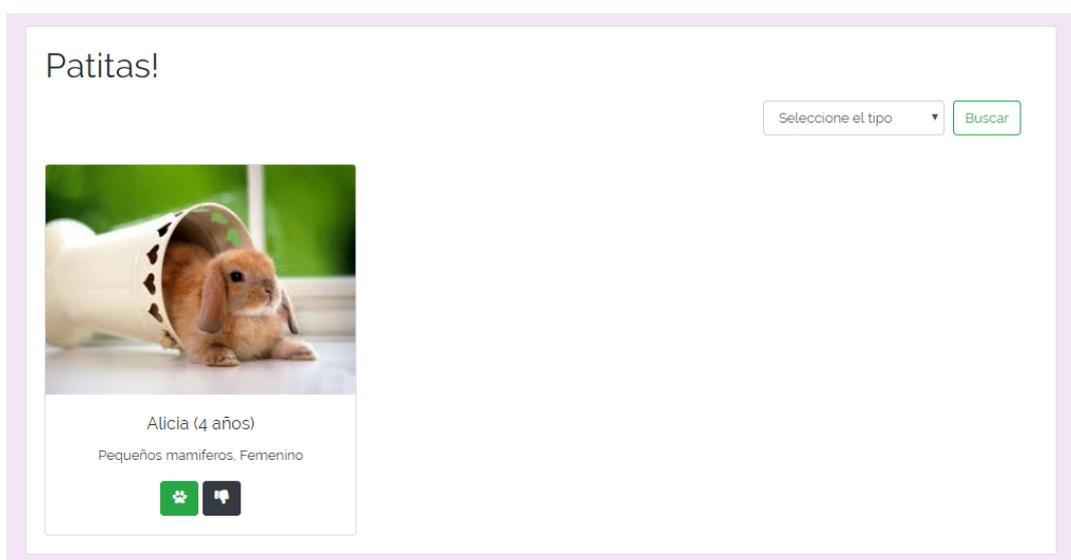


Figura 66. Demostración del paso 8 parte segunda.

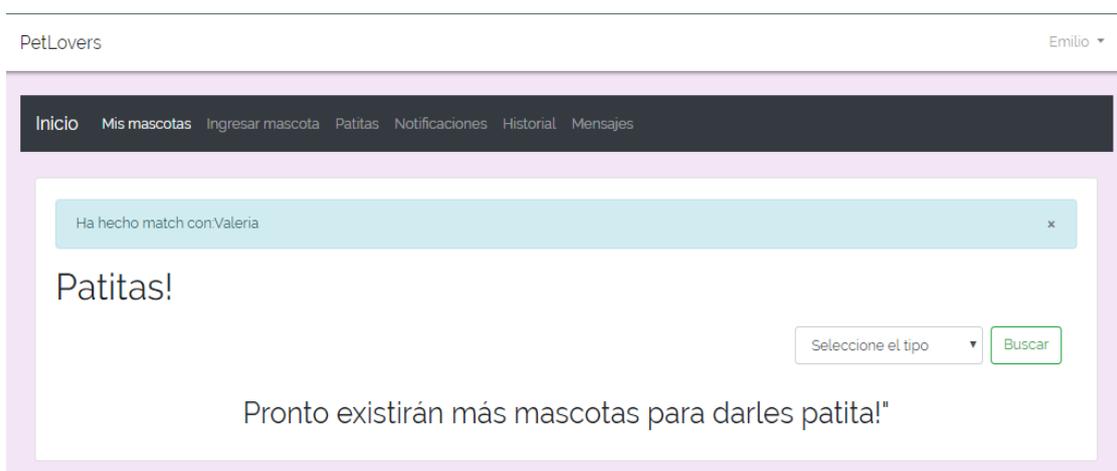


Figura 67. Demostración del paso 8 parte tercera.



Figura 68. Demostración del paso 9 parte primera.

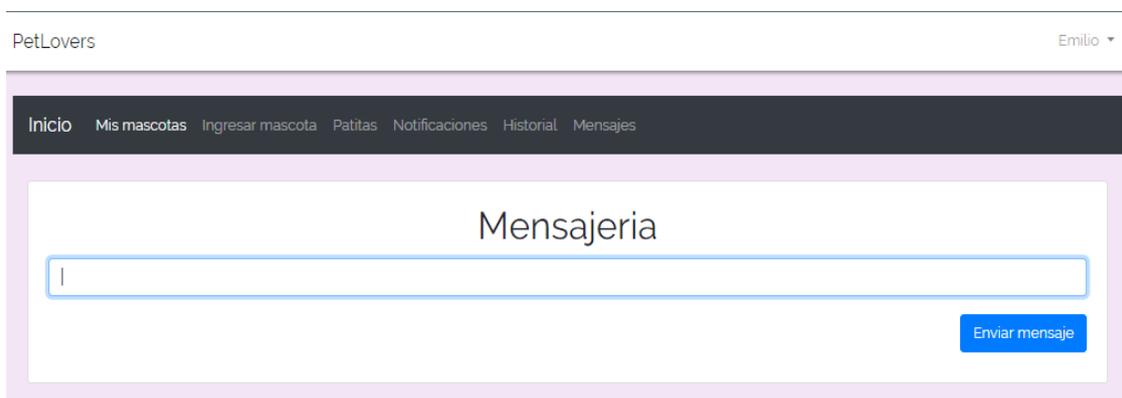


Figura 69. Demostración del paso 9 parte segunda.



Figura 70. Demostración del paso 10.

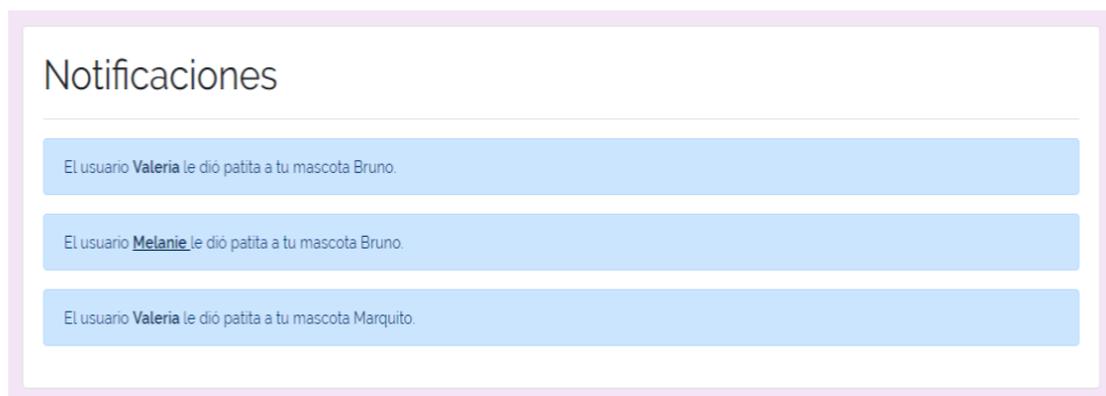


Figura 71. Demostración del paso 11 parte 1.

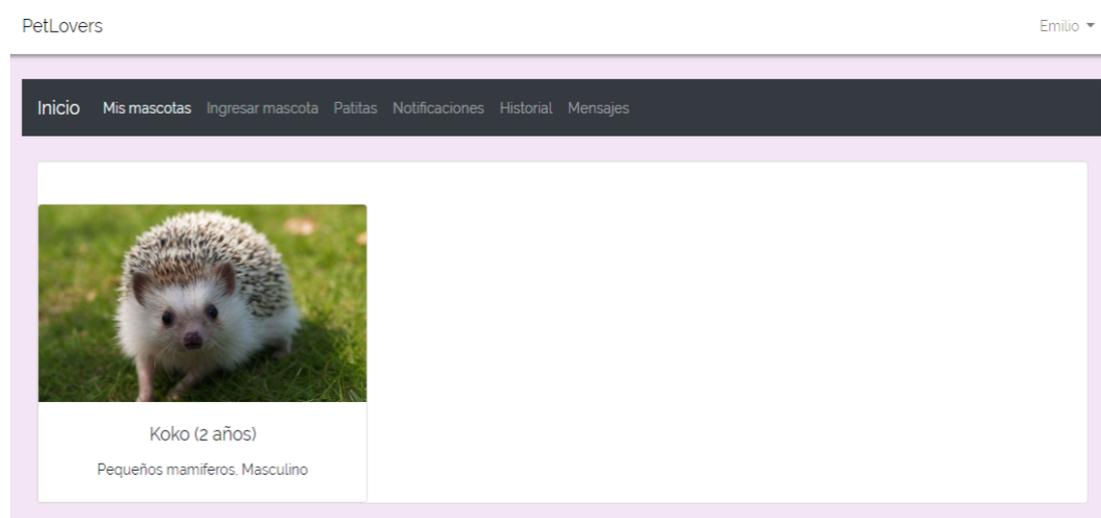


Figura 72. Demostración del paso 11 parte 2.

3.3 Pruebas Unitarias

Estas pruebas unitarias no se basan en los requerimientos del usuario, sino en lo implícito en ellas. Lo más importante al realizar estas pruebas es verificar que la solución no muestre errores o falle la aplicación si el usuario no realiza los pasos correctos (se entiende por pasos correctos a los pasos realizados en las pruebas de aceptación del cliente).

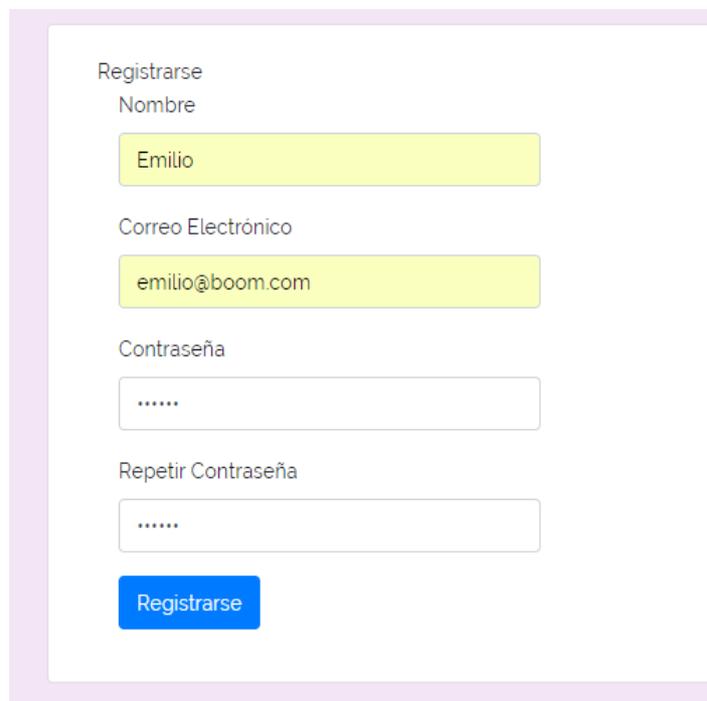
La tabla 5, describe un conjunto de casos en los cuales el usuario puede fallar al interactuar con el sistema y en la columna de estado si este error está controlado o no.

Tabla 5.

Pruebas unitarias.

Caso	Detalle	Estado
1	No permitir el registro de un usuario ya registrado anteriormente.	Controlado
2	Pedir los campos necesarios si no han sido llenados en el formulario de ingreso y registro.	Controlado
3	Pedir que se ingresen correctamente los datos en el formulario de ingreso y registro.	Controlado
4	No permitir la vista de ninguno de los elementos del menú hasta que el usuario se haya identificado.	Controlado
5	En caso de no existir mascotas ingresadas por parte del usuario, mostrar un mensaje describiendo que debe ingresar mascotas.	Controlado
6	En caso de no existir mascotas ingresadas por otros usuarios o ya no existan mascotas para reaccionar, mostrar un mensaje describiendo que pronto habrá mascotas para reaccionar.	Controlado
7	En caso de no existir notificaciones, mostrar un mensaje redactándolo.	Controlado
8	En caso de no existir historial, mostrar un mensaje redactándolo.	Controlado
9	En caso de no existir matches, mostrar un mensaje que diga que aún no tiene mensajes.	Controlado
10	Ingreso correcto de los datos en la base de datos	Controlado
11	Aún si una mascota es eliminada se conserva la conversación con los usuarios con los que se realizó el match.	Controlado
12	En caso de ingreso incorrecto de datos, mostrar informe de errores al usuario.	Controlado

En base a la tabla 5, las siguientes imágenes prueban su cumplimiento.



Registrarse

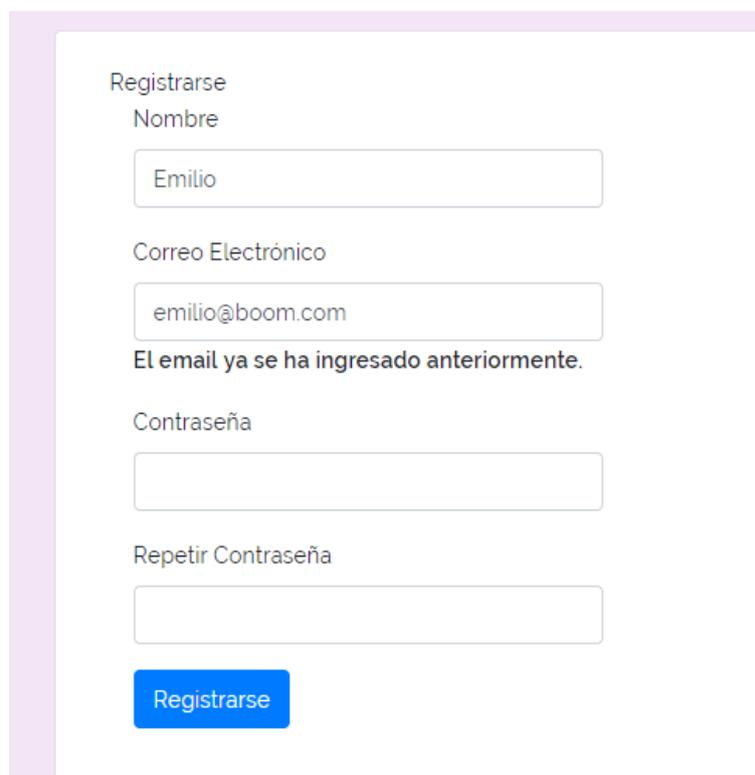
Nombre

Correo Electrónico

Contraseña

Repetir Contraseña

Figura 73. Demostración del caso 1 parte primera.



Registrarse

Nombre

Correo Electrónico

El email ya se ha ingresado anteriormente.

Contraseña

Repetir Contraseña

Figura 74. Demostración del caso 1 parte segunda.

Registrarse

Nombre

Correo Electrónico

Contraseña

Repetir Contraseña

Registrarse

Figura 75. Demostración del caso 2 parte primera.

Registrarse

Nombre

Correo Electrónico

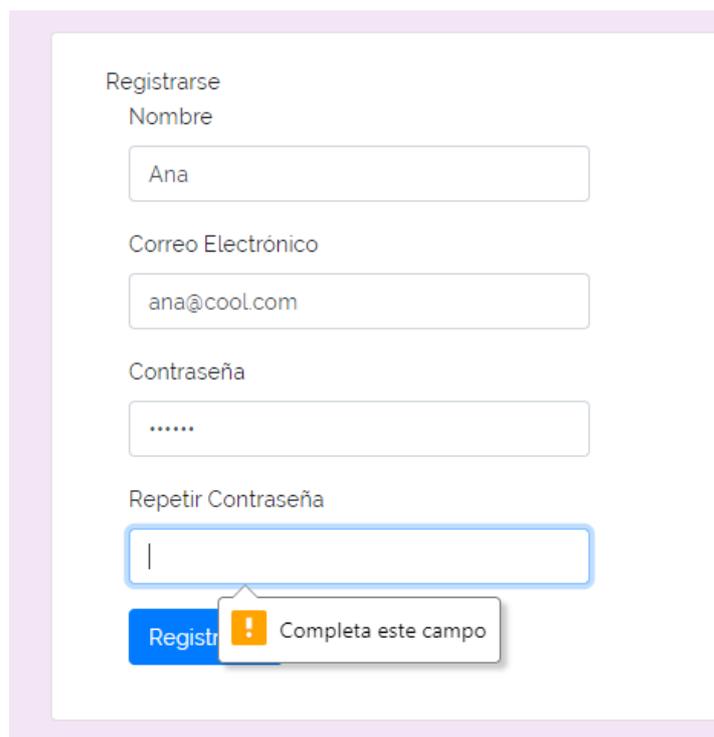
Contraseña

Completa este campo

Repetir Contraseña

Registrarse

Figura 76. Demostración del caso 2 parte segunda.



Registrarse

Nombre

Correo Electrónico

Contraseña

Repetir Contraseña

Registrarse

! Completa este campo

Figura 77. Demostración del caso 2 parte tercera.



Registrarse

Nombre

Correo Electrónico

! Incluye un signo "@" en la dirección de correo electrónico. La dirección "juan" no incluye el signo "@".

.....

Repetir Contraseña

Registrarse

Figura 78. Demostración del caso 3 parte primera.



Registrarse

Nombre

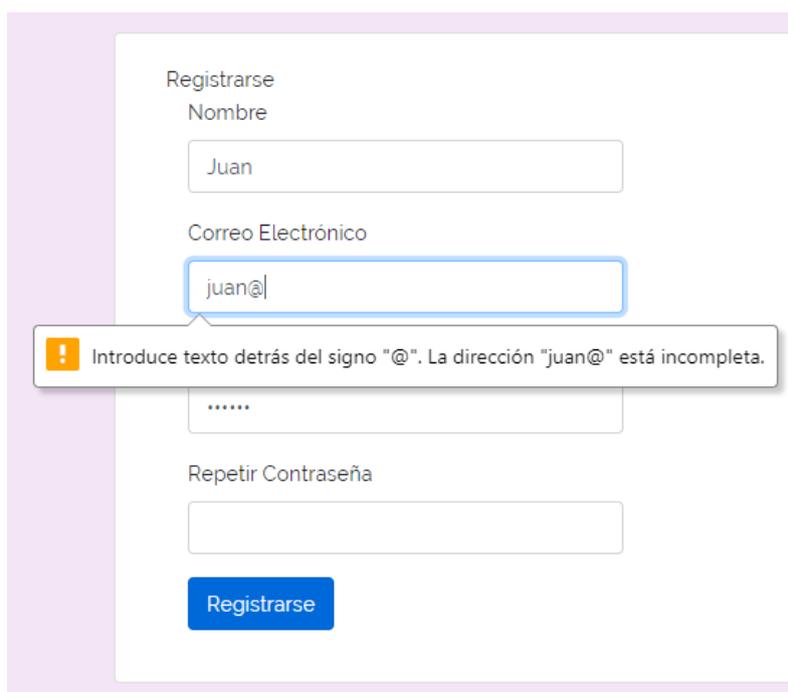
Correo Electrónico

Introduce texto detrás del signo "@". La dirección "juan@" está incompleta.

.....

Repetir Contraseña

Figura 79. Demostración del caso 3 parte segunda.



Registrarse

Nombre

Correo Electrónico

Introduce texto detrás del signo "@". La dirección "juan@" está incompleta.

.....

Repetir Contraseña

Figura 80. Demostración del caso 3 parte segunda.

Registrarse

Nombre

Correo Electrónico

Contraseña

Repetir Contraseña

Registrarse

Figura 81. Demostración del caso 3 parte tercera.

Registrarse

Nombre

Correo Electrónico

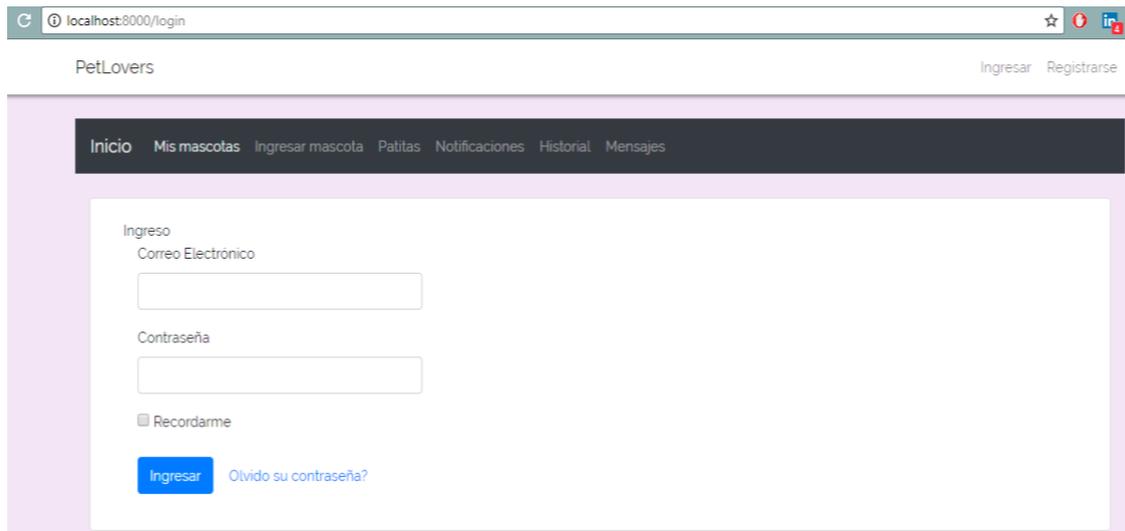
Contraseña

The password must be at least 6 characters.

Repetir Contraseña

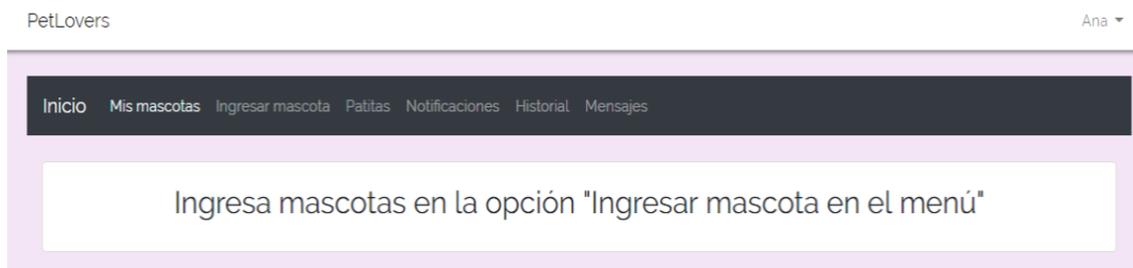
Registrarse

Figura 82. Demostración del caso 3 parte cuarta.



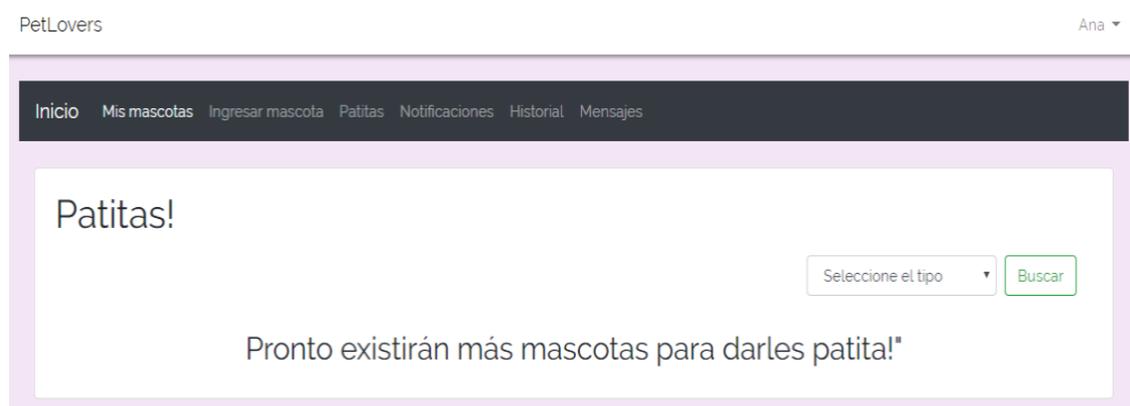
The screenshot shows a web browser window with the address bar displaying 'localhost:8000/login'. The page title is 'PetLovers'. In the top right corner, there are links for 'Ingresar' and 'Registrarse'. A dark navigation bar contains the following menu items: 'Inicio', 'Mis mascotas', 'Ingresar mascota', 'Patitas', 'Notificaciones', 'Historial', and 'Mensajes'. The main content area is titled 'Ingreso' and contains a login form with the following elements: a label 'Correo Electrónico' above a text input field, a label 'Contraseña' above a password input field, a checkbox labeled 'Recordarme', a blue 'Ingresar' button, and a link 'Olvido su contraseña?'.

Figura 83. Demostración del caso 4.



The screenshot shows the 'Ingresar mascota' page of the PetLovers application. The browser title is 'PetLovers' and the user name 'Ana' is visible in the top right. The navigation bar is identical to the previous screenshot. The main content area features a large white box with the text: 'Ingresa mascotas en la opción "Ingresar mascota en el menú"'. The text is centered and serves as an instruction for the user.

Figura 84. Demostración del caso 5.



The screenshot shows the 'Patitas' page of the PetLovers application. The browser title is 'PetLovers' and the user name 'Ana' is visible in the top right. The navigation bar is identical to the previous screenshots. The main content area features the heading 'Patitas!' followed by a search form. The search form includes a dropdown menu labeled 'Seleccione el tipo' and a green 'Buscar' button. Below the search form, the text reads: 'Pronto existirán más mascotas para darles patita!'. The text is centered and serves as a message to the user.

Figura 85. Demostración del caso 6.

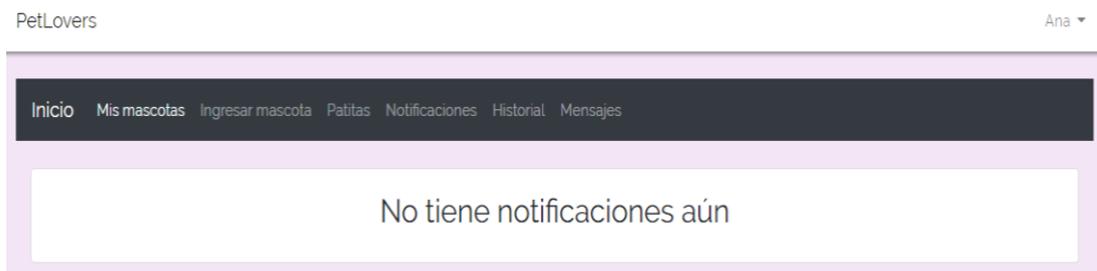


Figura 86. Demostración del caso 7.

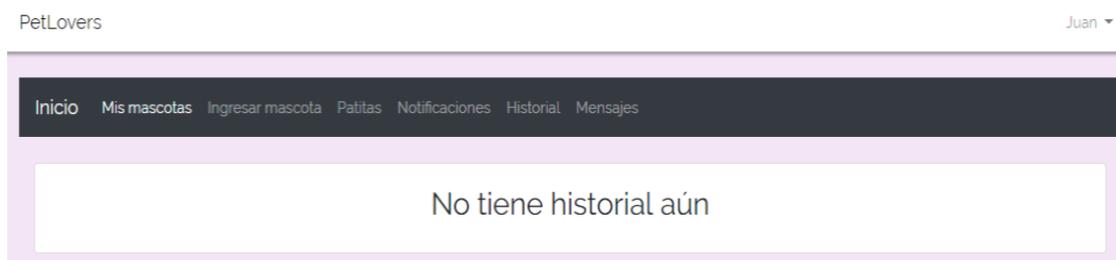


Figura 87. Demostración del caso 8.



Figura 88. Demostración del caso 9.



Figura 89. Demostración del caso 10 parte primera.



	id	humano_id	raza	nombre	edad	sexo	foto
<input type="checkbox"/>	1	1	Perro	Adrian II	1	Masculino	/images/pets/1_1528693176.jpg
<input type="checkbox"/>	2	2	Reptiles y anfibios	Sra. Puma	8	Femenino	/images/pets/2_1528693347.jpg
<input type="checkbox"/>	3	3	Gato	Meepo	3	Masculino	/images/pets/3_1528693419.jpg
<input type="checkbox"/>	4	4	Peces	Nina	2	Femenino	/images/pets/4_1528693476.jpg
<input type="checkbox"/>	5	5	Pequeños mamíferos	Bruno	7	Masculino	/images/pets/5_1528724938.jpg
<input type="checkbox"/>	6	3	Pequeños mamíferos	Alicia	4	Femenino	/images/pets/3_1528725974.jpg

Figura 90. Demostración del caso 10 parte segunda.

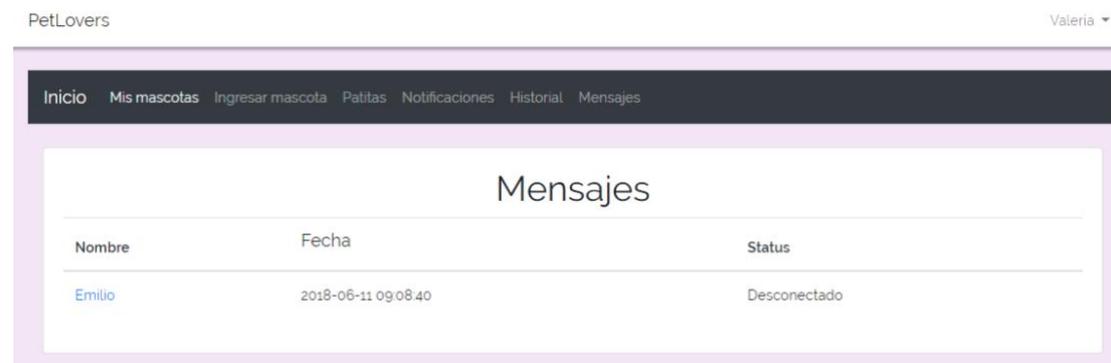


Figura 91. Demostración del caso 11 parte primera.

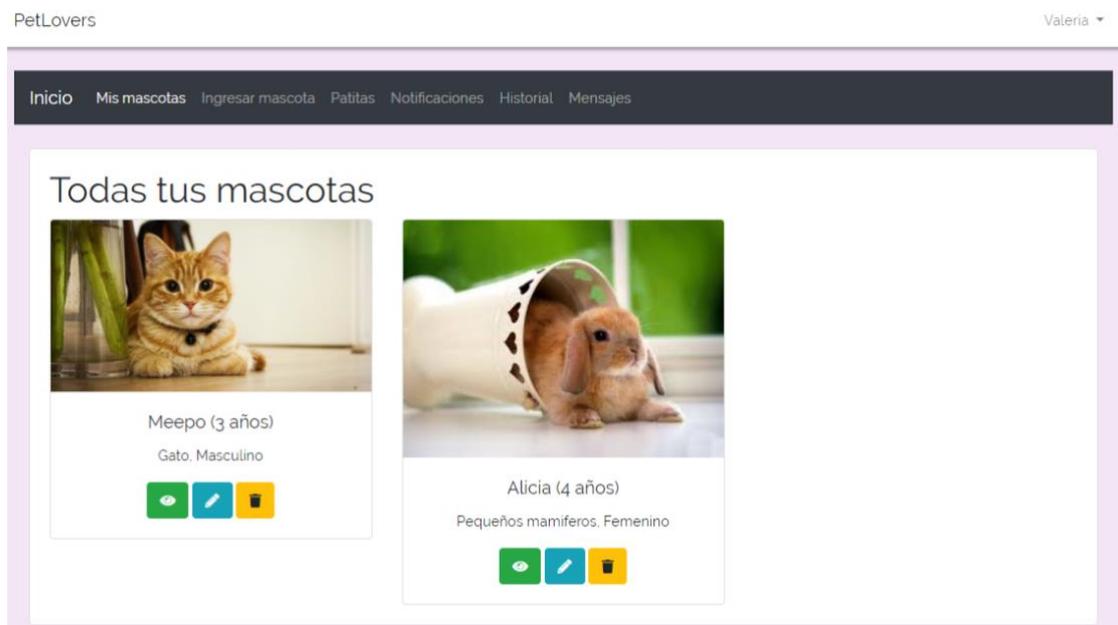


Figura 92. Demostración del caso 11 parte segunda.

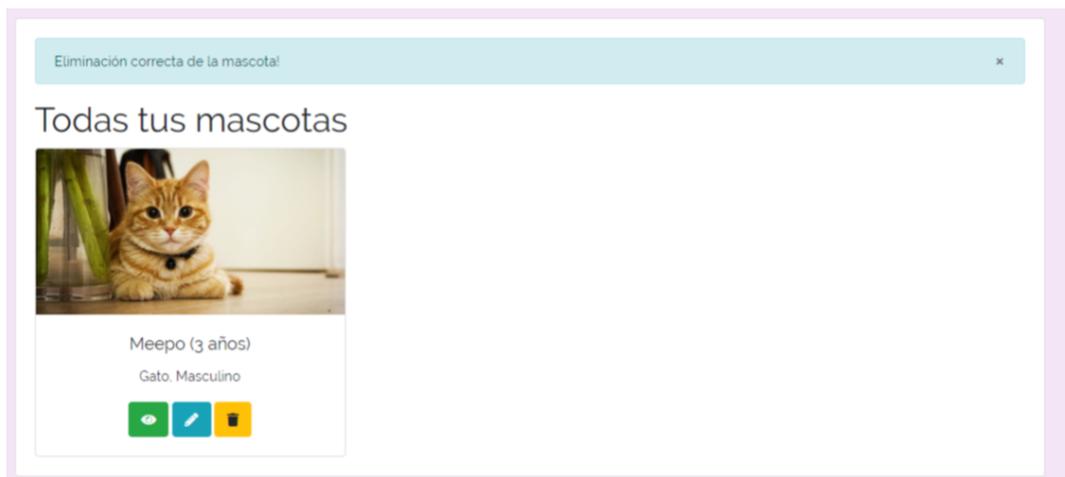


Figura 93. Demostración del caso 11 parte tercera.



Figura 94. Demostración del caso 11 parte cuarta.

Ingresar mascota

- El campo nombre es requerido.
- El campo edad es requerido.
- El campo foto es requerido.

Tipo

Nombre

Edad

Sexo

Foto Ningún archivo seleccionado

Figura 95. Demostración del caso 12.

4. Conclusiones y Recomendaciones

4.1 Conclusiones

Existe una variedad de herramientas y productos de software que intentan tratar la relación humano-animal. Por ejemplo, Mascotea es una página que abarca varias funcionalidades para las mascotas incluida una red social o Twindog que posee la función de match, pero solo es dirigida a canes. Sin embargo, la propuesta de solución planteada tiene un enfoque más centralizado en la estimulación de los vínculos afectivos humano-animal ya que se producen las interacciones por afinidad y como consecuencia de una respuesta positiva a estas interacciones se forma un match entre usuarios, desbloqueando así un canal de comunicación instantánea para ambos.

Las empresas hoy en día requieren el cumplimiento de calidad y bajos costos en el menor tiempo posible. Las metodologías ágiles ayudan a que esto sea posible permitiendo la entrega temprana de productos, entre ellas, se escogió el framework scrum por ser la más destacada.

Desarrollar un proyecto personal o comercial trae muchísimos desafíos. Para este en particular, se escogió el framework Laravel por su alta popularidad, la comunidad y documentación amplia, útil para aquellos usuarios que comienzan a desarrollar aplicaciones Web.

El propósito de utilizar Laravel se debe a que es de código abierto, se destaca también la utilización del patrón MVC, la inyección de dependencias y su consola Artisan. Permite extender plantillas creadas y secciones en otras vistas en las cuales también son accesibles las variables y la posibilidad de utilizar código PHP.

Con respecto a la metodología Scrum, los requerimientos siempre han sido parte del desarrollo de software, pero gracias al product backlog se tiene tanto la estimación de esfuerzo como la importancia de cada historia de usuario, haciendo que sea más fácil para el desarrollador comprender cómo serán formados los respectivos sprints backlogs y de esa manera ofrecer

incrementos operativos con las condiciones necesarias para ser entregadas al cliente final.

Es imposible desarrollar algo sin comprender las herramientas para hacerlo. El marco teórico fue la base para el desarrollo de la solución. Tanto el patrón MVC que marco la manera en que las capas de la solución se comunicarán consiguiendo una estructura estable, así como la tecnología Pusher y Vue.js que lograron la creación de mensajes en tiempo real con los cuales se logró la conexión entre los usuarios. La tecnología Pusher permite al desarrollador simplemente utilizar algo ya establecido y no tener que configurar los sockets de comunicación desde cero, brindando rapidez y soporte a la solución.

Las pruebas ayudan a que el software alcance estándares de calidad tanto en el área de programación y experiencia de usuario. La realización de estas ayuda a corregir errores que faltan pulir o analizar una forma de mejora que se podría desarrollar como actualización. Por ejemplo, en las pruebas realizadas se observó que muchas advertencias estaban en el idioma inglés y se cambió al español para mayor entendimiento.

4.2 Recomendaciones

Se pudiera agregar a la aplicación web actualizaciones varias, por ejemplo, más información al perfil de usuario, como fotografía y mayor descripción de este, para obtener más confianza en los usuarios.

Laravel está en constante actualización, en caso de existir una actualización del aplicativo web se recomienda revisar la documentación respectiva dependiendo de la versión de este proyecto.

Para que la solución funcione correctamente se necesita de conexión a internet y que se necesita del servidor de Pusher para que pueda existir la mensajería instantánea.

Otra actualización sugerida sería la realización de mostrar a las mascotas dependiendo de la ubicación, es decir, que se muestre a las mascotas cercanas al usuario. De esta manera se podrían conseguir amigos que se encuentren cerca del perímetro establecido.

Referencias

- Abellán, R. M. (2008). La terapia asistida por animales: una nueva perspectiva y línea de investigación en la atención a la diversidad. *Indivisa: Boletín de estudios e investigación*, (9), 117-146.
- Agarwal, A. (2017). *The Basics of SCRUM: A Simple Handbook to the Most Popular Agile Scrum Framework*.
- Bepuppy (2018). *Beppupy, pet social network*. Recuperado el 25 de mayo del 2018 de: <https://www.bepuppy.com/login>
- Bradshaw, J. (2017). *Behaviour of cats. The ethology of domestic animals*. 3rd ed. Wallingford: CAB International, 241-254.
- Bradshaw, J. (2017). *The Animals Among Us: How Pets Make Us Human*. 1rd ed. New York: Basic Books.
- Caba, M. (2003). Oxitocina: la hormona del amor materno.
- Cambridge University Press. (2018). pet, dictionary.cambridge.org. Recuperado el 18 de enero del 2018 de: <https://dictionary.cambridge.org/es/diccionario/ingles/pet>
- Calatrava, L. R., López, J. M. H., & López, F. (2015). Beneficios que aporta el animal como recurso para el mantenimiento de la calidad de vida tras la jubilación. *Calidad de vida, cuidadores e intervención para la mejora de la salud en el envejecimiento Volumen III*, 81.
- Clewley, R. (2016). *Animal Therapy: Healing Power of Pets*. (1.^a ed.). [versión electrónica]. Recuperado el 2 de enero del 2018 de: <https://books.google.com.ec/books?id=y4JADQAAQBAJ&printsec=frontcover&dq=Clewley+Healing+Power&hl=es&sa=X&ved=0ahUKEwi254736N7cAhUNtlkKHZ4jBGoQ6AEIKTAA#v=onepage&q=Clewley%20Healing%20Power&f=false>
- Cobo, Á. (2005). *PHP y MySQL: Tecnología para el desarrollo de aplicaciones web*. Ediciones Díaz de Santos.

- Datemypet (2018). Sitio de citas para adoradores de los animales. Recuperado el 25 de mayo del 2018 de: <https://www.datemypet.com/>
- El ciudadano (2017). 53 millones de dólares genera el mercado de los alimentos para mascotas. Recuperado el 14 de enero del 2018 de de: <http://www.elciudadano.gob.ec/53-millones-de-dolares-genera-el-mercado-delos-alimentos-para-mascotas/>.
- El Comercio (s.f). Estudio demuestra beneficios para los adultos mayores de tener una mascota. Recuperado el 14 de enero del 2018 de: <http://www.elcomercio.com/narices-frias/estudio-beneficios-mascotas-perros-adultosmayores.html>
- El Comercio (2014). Dueños de mascotas dedican más de dos horas semanales en sus cuidados. Recuperado el 14 de enero del 2018 de: <https://www.elcomercio.com/tendencias/duenos-de-mascotas-dedican-mas.html>
- El Comercio (s.f). Estudio demuestra beneficios para los adultos mayores de tener una mascota. Recuperado el 14 de enero del 2018 de: <http://www.elcomercio.com/narices-frias/estudio-beneficios-mascotas-perros-adultosmayores.html>
- Facepets (2018). Facepets – Red social de mascotas. Recuperado el 25 de mayo del 2018 de: <http://www.facepets.es/index.php>
- Gauchat, J. D. (2012). El gran libro de HTML5, CSS3 y Javascript. Barcelona: Marcombo.
- Gómez, L. F., Atehortua, C. G., & Padilla, S. C. O. (2007). La influencia de las mascotas en la vida humana. *Revista Colombiana de Ciencias Pecuarias*, 20(3), 377-386.
- Gutiérrez, G., Granados, D. R., & Piar, N. (2007). Interacciones humano-animal: características e implicaciones para el bienestar de los humanos. *Revista colombiana de psicología*.

- Gutiérrez, J. J. (2014). ¿Qué es un framework web? Recuperado el 12 de marzo del 2018 de: http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.Pdf
- Herzog, H. A. (2014). *Biology, culture, and the origins of pet-keeping*. *Animal Behavior and Cognition*, 1(3), 296-308.
- Monte, J. (2017). *Implantar SCRUM con éxito*. 1rd ed. Barcelona: UOC, 21-121.
- Martín, F. J., & Graciani, D. C. (2008). *PHP: Lenguaje de programación*. Departamento de Ciencias de la Computación e Inteligencia Artificial. Sevilla: Universidad de Sevilla.
- Mascotea (2018). Mascotea, mucho más que una red social de mascotas para perros, gatos, etc. Recuperado el 25 de mayo del 2018 de: <http://www.mascotea.net/es/>
- MySQL (2018). Reference Manual:1.3.1. What is MySQL? Recuperado el 21 de febrero del 2018 de: <https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html>
- Pichot, T., & Coulter, M. (2007). *Animal-assisted brief therapy: A solution-focused approach*. New York: Routledge.
- PHP (2018). ¿Qué es PHP? Recuperado el 17 de abril del 2018 de: <http://php.net/manual/es/intro-what-is.php>
- Puhser (2017). Documentación de Pusher. Recuperado el 5 de marzo del 2018 de: <https://pusher.com/docs>
- Puhser Community Github (2017). *What is Pusher?* Recuperado el 12 de febrero del 2018 de: <https://pusher-community.github.io/real-time-laravel/index.html>.
- Schwaber, K., & Beedle, M. (2002). *Agile software development with Scrum (Vol. 1)*. Upper Saddle River: Prentice Hall.
- Serpell, J. A., & Paul, E. S. (2011). *Pets in the Family: An Evolutionary*. *The Oxford handbook of evolutionary family psychology*, 297.

- Sierra, F., Acosta, J., Ariza, J., & Salas, M. (2017). Estudio y análisis de los framework en php basados en el modelo vista controlador para el desarrollo de software orientado a la web. *Revista Investigación y Desarrollo en TIC*, 4(2).
- Social Animals (2018). Bienvenido a Social Animals, la primera red social para ti y tus mascotas. Recuperado el 25 de febrero del 2018 de: <https://www.socialanimals.es/>
- Takeuchi, H., & Nonaka, I. (1986). *The new new product development game*. *Harvard business review*, 64(1), 137-146.
- Tindog (2018). Tindog, encuentra a otros perros y sus dueños cerca de ti. Recuperado el 15 de abril del 2018 de: <http://www.twindog.co/>
- Tucker, A. (2017). *Un león en el sofá: Cómo los gatos domésticos nos domesticaron y se apoderaron del mundo*. La Esfera de los Libros.
- Videla, M. D., & Olarte, M. A. (2016). Animales de compañía, personalidad humana y los beneficios percibidos por los custodios. *PSIENCIA. Revista Latinoamericana de Ciencia Psicológica*, 8(2).
- Villalta Gil, V., & Ochoa Güerre, S. (2007). La terapia facilitada por animales de compañía como programa de rehabilitación adjunto para personas con diagnóstico de esquizofrenia crónica. *Papeles del psicólogo*, 28(1).
- Vivapets (2018). La comunidad de mascotas. Recuperado el 28 de febrero del 2018 de: <https://www.vivapets.es/>
- Vue (2017). Guía de Vue, Introducción. Recuperado el 23 de enero del 2018 de sitio web: <https://vuejs.org/v2/guide/>
- Yummypets (2018). Yummypets, *the social network, is a community dedicated to pet lovers*. Recuperado el 25 de mayo del 2018 de: <https://www.yummypets.com/landing>.

Zamarra, M. P. (2002). Terapia asistida por animales de compañía. Bienestar para el ser humano. Temas de hoy. Marzo. p 143-49. Association for the Study of Diabetes (EASD). Eur Heart J, 28, 88-136.

